AD628956

CONCATENATED CODES

G. DAVID FORNEY, JR.

TECHNICAL REPORT 440

DECEMBER 1, 1965

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
CAMBRIDGE, MASSACHUSETTS

The Research Laboratory of Electronics is an interdepartmental laboratory in which faculty members and graduate students from numerous academic departments conduct research.

The research reported in this document was made possible in part by support extended the Massachusetts Institute of Technology, Research Laboratory of Electronics, by the JOINT SERVICES ELECTRONICS PROGRAMS (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract No. DA36-039-AMC-03200(E); additional support was received from the National Science Foundation (Grant GP-2495), the National Institutes of Health (Grant MH-04737-05), and the National Aeronautics and Space Administration (Grants NsG-334 and NsG-496).

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 440 December 1, 1965

CONCATENATED CODES

G. David Forney, Jr.

Submitted to the Department of Electrical Engineering, M. I. T., March 31, 1965, in partial fulfillment of the requirements for the degree of Doctor of Science.

(Manuscript received April 2, 1965)

Abstract

Concatenation is a method of building long codes out of shorter ones; it attempts to meet the problem of decoding complexity by breaking the required computation into manageable segments. We present theoretical and computational results bearing on the efficiency and complexity of concatenated codes; the major theoretical results are the following:

1. Concatenation of an arbitrarily large number of codes can yield a probability of error that decreases exponentially with the over-all block length, while the decoding complexity increases only algebraically; and

2. Concatenation of a finite number of codes yields an error exponent that is inferior to that attainable with a single stage, but is nonzero at all rates below capacity.

Computations support these theoretical results, and also give insight into the relationship between modulation and coding.

This approach illuminates the special power and usefulness of the class of Reed-Solomon codes. We give an original presentation of their structure and properties, from which we derive the properties of all BCH codes; we determine their weight distribution, and consider in detail the implementation of their decoding algorithm, which we have extended to correct both erasures and errors and have otherwise improved. We show that on a particularly suitable channel, RS codes can achieve the performance specified by the coding theorem.

Finally, we present a generalization of the use of erasures in minimum-distance decoding, and discuss the appropriate decoding techniques, which constitute an interesting hybrid between decoding and detection.

TABLE OF CONTENTS

CONTENTS

I. INTRODUCTION

It is almost twenty years since Shannon[1] announced the coding theorem. The promise of that theorem was great: a probability of error exponentially small in the block length at any information rate below channel capacity. Finding a way of implementing even moderately long codes, however, proved much more difficult than was imagined at first. Only recently, in fact, have there been invented codes and decoding methods powerful enough to improve communication system performance significantly yet simple enough to be attractive to build.[2-4]

The work described here is an approach to the problem of coding and decoding complexity. It is based on the premise that we may not mind using codes from 10 to 100 times longer than the coding theorem proves to be sufficient, if, by so doing, we arrive at a code that we can implement. The idea is basically that used in designing any large system: break the system down into subsystems of a size that can be handled, which can be joined together to perform the functions of the large system. A system so designed may be suboptimal in comparison with a single system designed all of a piece, but as long as the nonoptimalities are not crippling, the segmented approach may be the preferred engineering solution.

1.1 CODING THEOREM FOR DISCRETE MEMORYLESS CHANNELS

The coding theorem is an existence theorem. It applies to many types of channels, but generally it is similar to the coding theorem for block codes on discrete memoryless channels, which will now be stated in its most modern form.[5]

A discrete memoryless channel has I inputs x_i, J outputs y_j, and a characteristic transition probability matrix $p_{ji} \equiv \Pr(y_j/x_i)$. On each use of the channel, one of the inputs x_i is selected by the transmitter. The conditional probability that the receiver then observes the output y_j is p_{ji}; the memorylessness of the channel implies that these probabilities are the same for each transmission, regardless of what happened on any other transmission. A code word of length N for such a channel then consists of a sequence of N symbols, each of which comes from an I-symbol alphabet and denotes one of the I channel inputs; upon the transmission of such a word, a received word of length N becomes available to the receiver, where now the received symbols are from a J-symbol alphabet and correspond to the channel outputs. A block code of length N and rate R (nats) consists of e^{NR} code words of length N. Clearly $e^{NR} \leq I^N$; sometimes we shall use the dimensionless rate r, $0 \leq r \leq 1$, defined by $I^{rN} = e^{NR}$ or $R = r \ln I$.

The problem of the receiver is generally to decide which of the e^{NR} code words was sent, given the received word; a wrong choice we call an error. We shall assume that all code words are equally likely; then the optimal strategy for the receiver in principle, though rarely feasible, is to compute the probability of getting the received word, given each code word, and to choose that code word for which this probability is greatest; this strategy is called maximum-likelihood decoding. The coding theorem then

asserts that there exists a block code of length N and rate R such that with maximum-likelihood decoding the probability of decoding error is bounded by

$$\Pr(e) \leq e^{-NE(R)},$$

where E(R), the error exponent, is characteristic of the channel, and is positive for all rates less than C, called capacity.

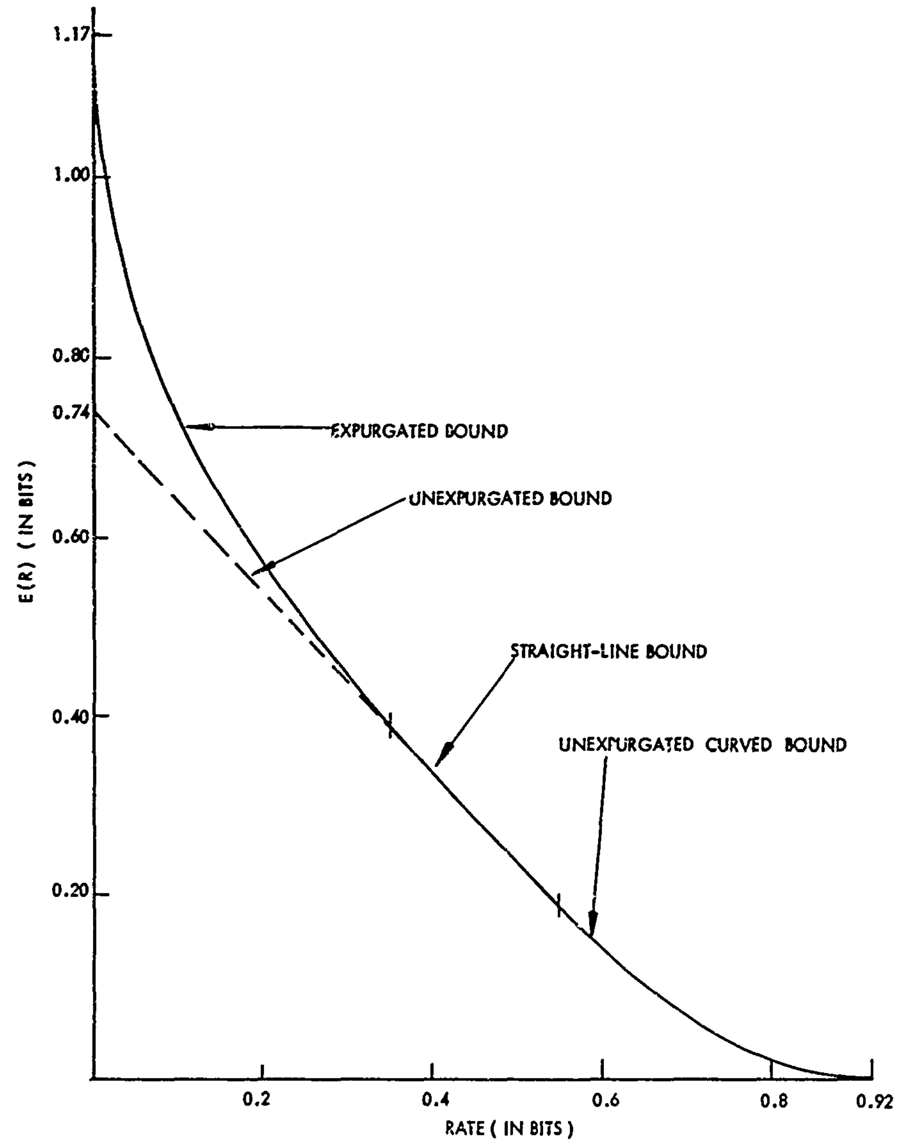

Fig. 1. E(R) curve for BSC with $p = .01$.

Figure 1 shows the error exponent for the binary symmetric channel whose cross-over probability is .01 — that is, the discrete memoryless channel with transition probability matrix $p_{11} = p_{22} = .99$, $p_{12} = p_{21} = .01$. As is typical, this curve has three segments: two convex curves joined by a straight-line segment of slope −1. Gallager[5] has shown that the high-rate curved segment and the straight-line part of the error exponent are given by

$$E(R) = \max_{\substack{0<\rho\leq 1 \\ \vec{P}}} \{E_o(\vec{P},\rho)-\rho R\}$$

where

$$E_o(\vec{P},\rho) \equiv -\ln \sum_{j=1}^{J} \left[\sum_{i=1}^{I} P_i P_{ji}^{1/(1+\rho)} \right]^{1+\rho},$$

$\vec{P}$ being any I-dimensional vector of probabilities P_i; this is called the unexpurgated error exponent, in deference to the fact that a certain purge of poor code words is involved in the argument which yields the low-rate curved segment, or expurgated error exponent. An analogous formula exists for the exponent when the inputs and outputs form continuous rather than discrete sets. It should be mentioned that a lower bound to Pr(e) is known which shows that in the range of the high-rate curved segment, this exponent is the true one, in the sense that there is no code which can attain $\Pr(e) \leq e^{-NE^*(R)}$ for $E^*(R) > E(R)$ and N arbitrarily large.

Thus for any rate less than capacity, the probability of error can be made to decrease exponentially with the block length. The deficiencies of the coding theorem are that it does not specify a particular code that achieves this performance, nor does it offer an attractive decoding method. The former deficiency is not grave, since the relatively easily implemented classes of linear codes[6] and convolutional codes[7] contain members satisfying the coding theorem. It has largely been the decoding problem that has stymied the application of codes to real systems, and it is this problem which concatenation attempts to meet.

1.2 CONCATENATION APPROACH

The idea behind concatenated codes is simple. Suppose we set up a coder and decoder for some channel; then the coder-channel-decoder chain can be considered from the outside as a superchannel with exp NR inputs (the code words), exp NR outputs (the decoder's guesses), and a transition probability matrix characterized by a high probability of getting the output corresponding to the correct input. If the original channel is memoryless, the superchannel must be also, if the code is not changed from block to block. It is now reasonable to think of designing a code for the superchannel of length n, dimensionless rate r, and with symbols from an e^{NR}-symbol alphabet. This done, we can abandon the fiction of the superchannel, and observe that we have created a code for the original channel of length nN, with $(e^{NR})^{Nr}$ code words, and therefore rate rR (nats). These ideas are illustrated in Fig. 2, where the two codes and their associated coders and decoders are labelled inner and outer, respectively.

By concatenating codes, we can achieve very long codes, capable of being decoded by two decoders suited to much shorter codes. We thus realize considerable savings in complexity, but at some sacrifice in performance. In Section V we shall find that this

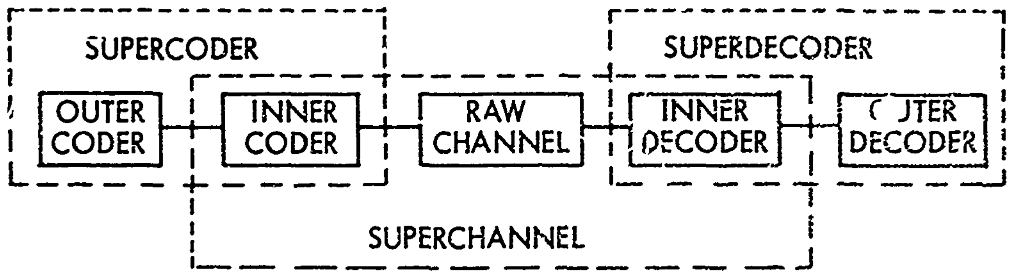

Fig. 2. Illustrating concatenation.

sacrifice comes in the magnitude of the attainable error exponent; however, we find that the attainable probability of error still decreases exponentially with block length for all rates less than capacity.

The outer code will always be one of a class of nonbinary BCH codes called Reed-Solomon[8] codes, first because these are the only general nonbinary codes known, and second, because they can be implemented relatively easily, both for coding and for decoding. But furthermore, we discover in Section V that under certain convenient suppositions about the superchannel, these codes are capable of matching the performance of the coding theorem. Because of their remarkable suitability for our application, we devote considerable time in Section III to development of their structure and properties, and in Section IV to the detailed exposition of their decoding algorithm.

1.3 MODULATION

The functions of any data terminal are commonly performed by a concatenation of devices; for example, a transmitting station might consist of an analog-to-digital converter, a coder, a modulator, and an antenna. Coding theory is normally concerned only with the coding stage, which typically accepts a stream of bits and delivers to the modulator a coded stream of symbols. Up to this point, only the efficient design of this stage has been considered, and in the sequel this concentration will largely continue, since this problem is most susceptible to analytical treatment.

By a raw channel, we mean whatever of the physical channel and associated terminal equipment are beyond our design control. It may happen that the channel already exists in such a form, say, with a certain kind of repeater, that it must be fed binary symbols, and in this case the raw channel is discrete. Sometimes, however, we have more freedom to choose the types of signals, the amount of bandwidth, or the amount of diversity to be used, and we must properly consider these questions together with coding to arrive at the most effective and economical signal design.

When we are thus free to select some parameters of the channel, the channel contemplated by algebraic coding theory, which, for one thing, has a fixed number of inputs and outputs, is no longer a useful model. A more general approach to communication

theory, usually described under the headings modulation theory, signal design, and detection theory, is then appropriate. Few general theoretical results are obtainable in these disciplines, which must largely be content with analyzing the performance of various interesting systems. Section VI reports the results of a computational search for coding schemes meeting certain standards of performance, where both discrete raw channels and channels permitting some choice of modulation are considered. This gives considerable insight into the relationship between modulation and coding. In particular it is shown that nonbinary modulation with relatively simple codes can be strikingly superior either to complicated modulation with no coding, or to binary modulation with complicated binary codes.

1.4 CHANNELS WITH MEMORY

Another reason for the infrequent use of codes in real communication systems has been that real channels are usually not memoryless. Typically, a channel will have long periods in which it is good, causing only scattered random errors, separated by short bad periods or bursts of noise. Statistical fluctuations having such an appearance will be observed even on a memoryless channel; the requirement of long codes imposed by the coding theorem may be interpreted as insuring that the channel be used for enough transmissions that the probability of a statistical fluctuation bad enough to cause an error is very small indeed. The coding theorem can be extended to channels with memory, but now the block lengths must generally be very much longer, so that the channel has time to run through all its tricks in a block length.

If a return channel from the receiver to the transmitter is available, it may be used to adapt the coding scheme at the transmitter to the type of noise currently being observed at the receiver, or to request retransmission of blocks which the receiver cannot decode.[9] Without such a feedback channel, if the loss of information during bursts is unacceptable, some variant of a technique called interlacing is usually envisioned.[10] In interlacing, the coder codes n blocks of length N at once, and then transmits the n first symbols, the n second symbols, and so forth through the n N^{th} symbols. At the receiver the blocks are unscrambled and decoded individually. It is clear that a burst of length $b \leq n$ can affect no more than one symbol in any block, so that if the memory time of the channel is of the order of n or less the received block of nN symbols will generally be decodable.

Concatenation obviously shares the burst-resistant properties of interlacing when the memory time of the channel is of the order of the inner code block length or less, for a burst then will usually affect no more than one or two symbols in the outer code, which will generally be quite correctable. Because of the difficulty of constructing adequate models of real channels with memory, it is difficult to pursue analysis of the burst resistance of concatenated codes, but it may be anticipated that this feature will prove useful in real applications.

1.5 CONCATENATING CONVOLUTIONAL CODES

We shall consider only block codes henceforth. The principles of concatenation are clearly applicable to any type of code. For example, a simple convolutional code with threshold decoding is capable of correcting scattered random errors, but when channel errors are too tightly bunched the decoder is thrown off stride for awhile, and until it becomes resynchronized causes a great many decoding errors. From the outside, such a channel appears to be an ideal bursty channel, in which errors do not occur at all except in the well-defined bursts. Very efficient codes are known for such channels, and could be used as outer codes. The reader will no doubt be able to conceive of other applications.

1.6 OUTLINE

This report consists of 6 largely self-sufficient sections, with two appendices. We anticipate that many readers will find that the material is arranged roughly in inverse order of interest. Therefore, we shall outiine the substance of each section and the connections between them.

Section II begins with an elaborate presentation of the concepts of minimum-distance decoding, which has two purposes: to acquaint the reader with the substance and utility of these concepts, and to lay the groundword for a generalization of the use of erasures in minimum-distance decoding. Though this generalization is an interesting hybrid between the techniques of detection and of decoding, it is not used subsequently.

Section III is an attempt to provide a fast, direct route for the reader of little background to an understanding of BCH codes and their properties. Emphasis is placed on the important nonbinary Reed-Solomon codes. Though the presentation is novel, the only new results concern the weight distribution of RS codes and the implementation of much shortened RS codes.

Section IV reports an extension of the Gorenstein-Zierler error-correcting algorithm for BCH codes so that both erasures and errors can be simultaneously corrected. Also, the final step in the GZ algorithm is substantially simplified. A close analysis of the complexity of implementing this algorithm with a computer concludes this section, and only the results of this analysis are used in the last two sections. Appendix A contains variants on this decoding algorithm of more restricted interest.

Section V contains our major theoretical results on the efficiency and complexity of concatenated codes, and Section VI reports the results of a computational program evaluating the performance of concatenated codes under a variety of specifications. The reader interested chiefly in the theoretical and practical properties of these codes will turn his attention first to Sections V and VI. Appendix B develops the formulas used in the computational program of Section VI.

II. MINIMUM-DISTANCE DECODING

We introduce here the concepts of distance and minimum-distance codes, and discuss how these concepts simplify decoding. We describe the use of erasures, and of a new generalization of erasures. Using the Chernoff bound, we discover the parameters of these schemes which maximize the probability of correct decoding; using the Gilbert bound, we compute the exponent of this probability for each of three minimum-distance decoding schemes over a few simple channels.

2.1 ERRORS-ONLY DECODING

In Section I we described how an inner code of length N and rate R could be concatenated with an outer code of length n and dimensionless rate r to yield a code of overall length nN and rate rR for some raw channel. Suppose now one of the e^{nNrR} words of this code is selected at random and transmitted — how do we decode what is received?

The optimum decoding rule remains what it always is when inputs are equally likely: the maximum-likelihood decoding rule. In this case, given a received sequence $\vec{r}$ of length nN, the rule would be to compute $\Pr(\vec{r}|\vec{f})$ for each of the e^{nNrR} code words $\vec{f}$.

The whole point of concatenation, however, is to break the decoding process into manageable segments, at the price of suboptimality. The basic simplification made possible by the concatenated structure of the code is that the inner decoder can decode (make a hard decision on) each received N-symbol sequence independently. In doing so, it is in effect discarding all information about the received N-symbol block except which of the e^{NR} inner code words was most likely, given that block. This preliminary processing enormously simplifies the task of the outer decoder, which is to make a final choice of one of the e^{nNrR} total code words.

Let $q = e^{NR}$. When the inner decoder makes a hard decision, the outer coder and decoder see effectively a q-input, q-output superchannel. We assume that the raw channel and thus the superchannel are memoryless. By a <u>symbol error</u> we shall mean the event in which any output but the one corresponding to the input actually transmitted is received. Normally, the probability of symbol error is low; it is then convenient to assume that all incorrect transmissions are equally probable — that is, to assume that the transition probability matrix of the superchannel is

$$p_{ji} = \begin{cases} \dfrac{p}{q-1}, & i \neq j \\ 1-p, & i = j \end{cases} \tag{1}$$

where p is the probability of decoding error in the inner decoder, hence of symbol error in the superchannel. We call a channel with such a transition probability matrix an <u>ideal superchannel</u> with q inputs and probability of error p.

Recall that the maximum-likelihood rule, given $\vec{r}$, is to choose the input sequence $\vec{f}$ for which the probability of receiving $\vec{r}$, given $\vec{f}$, is greatest. When

the channel is memoryless,

$$\Pr(\vec{r}|\vec{f}) = \prod_{i=1}^{n} \Pr(r_i|f_i).$$

But since log x is a monotonic function of x, this is equivalent to maximizing

$$\log \prod_{i=1}^{n} \Pr(r_i|f_i) = \sum_{i=1}^{n} \log \Pr(r_i|f_i). \tag{2}$$

Now for an ideal superchannel, substituting Eqs. 1 in Eq. 2, we want to maximize

$$\sum_{i=1}^{n} a'(r_i, f_i), \tag{3}$$

where

$$a'(r_i, f_i) \equiv \begin{cases} \log (1-p), & r_i = f_i \\ \log \left(\frac{p}{q-1}\right), & r_i \neq f_i. \end{cases}$$

Define the Hamming weight $a(r_i, f_i)$ by

$$a(r_i, f_i) \equiv \begin{cases} 0, & r_i = f_i \\ 1, & r_i \neq f_i. \end{cases} \tag{4}$$

Since

$$a'(r_i, f_i) = \log (1-p) + \left[\log \frac{p}{(q-1)(1-p)}\right] a(r_i, f_i),$$

maximizing Eq. 3 is equivalent to maximizing

$$n \log (1-p) + \left[\log \frac{p}{(q-1)(1-p)}\right] \sum_{i=1}^{n} a(r_i, f_i).$$

Under the assumption $p/(q-1) \leq (1-p)$, this is equivalent to minimizing

$$d_H(\vec{r}, \vec{f}) \equiv \sum_{i=1}^{n} a(r_i, f_i). \tag{5}$$

$d_H(\vec{r}, \vec{f})$ is called the Hamming distance[11] between $\vec{r}$ and $\vec{f}$, and is simply the number of places in which they differ. For an ideal superchannel, the maximum-likelihood decoding rule is therefore to choose that code word which is closest to the received word in Hamming distance.

Although this distance has been defined between a received word and a code word, there is no difficulty in extending the definition to apply between any two code words. We then define the minimum distance of a code as the minimum Hamming distance between any two words in the code.

A code with large minimum distance is desirable on two counts. First, as we shall now show, it insures that all combinations of less than or equal to a certain number t of symbol errors in n uses of the channel will be correctable. For, suppose $\vec{f}$ is sent and t symbol errors occur, so that $r_i \neq f_i$ in t places. Then from Eq. 5

$$d_H(\vec{r},\vec{f}) = t. \tag{6}$$

Take some other code word $\vec{g}$. We separate the places into three disjoint sets, such that

$$i \in \begin{cases} S_o & \text{if } f_i = g_i \\ S_c & \text{if } f_i \neq g_i \text{ and } r_i = f_i \\ S_e & \text{if } f_i \neq g_i \text{ and } r_i \neq f_i. \end{cases} \tag{7}$$

We note that the set S_e can have no more than t elements. Now the distance between r and $\vec{g}$,

$$d_H(\vec{r},\vec{g}) = \sum_{i=1}^{n} a(r_i,g_i)$$

$$= \sum_{i\in S_o} a(r_i,g_i) + \sum_{i\in S_c} a(r_i,g_i) + \sum_{i\in S_e} a(r_i,g_i), \tag{8}$$

can be lower-bounded by use of the relations

$$\begin{aligned} a(r_i,g_i) &\geq a(g_i,f_i) = 0, && i \in S_o \\ a(r_i,g_i) &= a(g_i,f_i) = 1, && i \in S_c \\ a(r_i,g_i) &\geq a(g_i,f_i) - 1 = 0, && i \in S_e \end{aligned} \tag{9}$$

Here, besides Eqs. 7, we have used $a \geq 0$ and the fact that for $i \in S_c$, $r_i \neq g_i$. Substituting (9) in (8) yields

$$d_H(\vec{r},\vec{g}) \geq d_H(\vec{g},\vec{f}) - |S_e| \geq d - t. \tag{10}$$

Here, we have defined $|S_e|$ as the number of elements in S_e and used the fact that $d_H(\vec{g},\vec{f}) \geq d$ if $\vec{g}$ and $\vec{f}$ are different words in a code with minimum distance d. By combining (6) and (10) we have proved that

$$d_H(\vec{r},\vec{f}) < d_H(\vec{r},\vec{g}) \quad \text{if } 2t < d. \tag{11}$$

In other words, if t_o is the largest integer such that $2t_o < d$, it is impossible for any combination of t_o or fewer symbol errors to cause the received word to be closer to any other code word than to the sent word. Therefore no decoding error will occur.

Another virtue of a large minimum distance follows from reinterpreting the argument above. Suppose we hypothesize the transmission of a particular code word; given the received word, this hypothesis implies the occurrence of a particular sequence of errors. If this sequence is such that the Hamming distance criterion of Eq. 11 is satisfied, then we say that the received word is <u>within the minimum distance</u> of that code word. (This may seem an unnecessarily elaborate way of expressing this concept, but, as in this whole development, we are taking great pains now so that the generalizations of the next two sections will follow easily.) Furthermore, the preceding argument shows that there can be no more than one code word within the minimum distance of the received word. Therefore, if by some means the decoder generates a code word that it discovers to be within the minimum distance of the received word, it can without further ado announce that word as its maximum-likelihood choice, since it knows that it is impossible that there be any other code word as close or closer to the received word. This property is the basis for a number[12-15] of clever decoding schemes proposed recently, and will be used in the generalized minimum-distance decoding of section 2.3.

A final simplification that is frequently made is to set the outer decoder to decode only when there is a code word within the minimum distance of the received word. Such a scheme we call <u>errors-only decoding</u>. There will of course in general be received words beyond the minimum distance from all code words, and on such words an errors-only decoder will fail. Normally, a decoding failure is not distinguished from a decoding error, although it is detectable while an error is not.

2.2 DELETIONS-AND-ERRORS DECODING

The simplifications of the previous section were bought, we recall, at the price of denying to the outer decoder all information about what the inner decoder received except which of the inner code words was most probable, given that reception. In this and the following section we investigate techniques of relaying somewhat more information to the outer decoder, hopefully without greatly complicating its task. These techniques are generalizations of errors-only decoding, and will be developed in the framework that has been introduced.

We continue to require the inner decoder to make a hard decision about which code word was sent. We now permit it to send along with its guess some indication of how reliable it considers its guess to be. In the simplest such strategy, the inner decoder indicates either that its guess is fully reliable or completely unreliable; the latter event is called a <u>deletion</u> or <u>erasure</u>. The inner decoder normally would delete whenever the evidence of the received word did not clearly indicate which code word was sent; also, a decoding failure, which can occur in errors-only decoding, would be treated as a deletion, with some arbitrary word chosen as the guess.

In order to make use of this reliability information in minimum distance decoding, we define the Elias weight by

$$b(r_i, f_i) \equiv \begin{cases} 0, & r_i \text{ reliable and } r_i = f_i \\ \beta, & r_i \text{ erased} \\ 1, & r_i \text{ reliable and } r_i \neq f_i \end{cases} \tag{12}$$

where β is an arbitrary number between zero and one. Then the Elias distance[16] between a received word $\vec{r}$ and a code word $\vec{f}$ is defined as

$$d_E(\vec{r}, \vec{f}) \equiv \sum_{i=1}^{n} b(r_i, f_i). \tag{13}$$

Note that Elias distance is not defined between two code words.

We shall let our decoding rule be to choose that code word which is closest in Elias distance to the received word. Let us then suppose that some word $\vec{f}$ from a code of minimum (Hamming) distance d is transmitted, and in the n transmissions (i) s deletions occur, and (ii) t of the symbols classed as reliable are actually incorrect. Then

$$d_E(\vec{r}, \vec{f}) = t + \beta s. \tag{14}$$

Take some other code word $\vec{g}$. We separate the places into disjoint sets such that

$$i \in \begin{cases} S_o & \text{if } f_i = g_i \\ S_c & \text{if } f_i \neq g_i, \quad r_i = f_i, \ r_i \text{ reliable} \\ S_d & \text{if } f_i \neq g_i, \quad r_i \text{ deleted} \\ S_e & \text{if } f_i \neq g_i, \quad r_i \neq f_i, \ r_i \text{ reliable} \end{cases} \tag{15}$$

Note that

$$|S_e| \leq t$$

and (16)

$$|S_d| \leq s.$$

Now the distance between $\vec{r}$ and $\vec{g}$ can be lower-bounded by the relations

$$\begin{aligned} b(r_i, g_i) &\geq a(g_i, f_i) = 0, & i \in S_o \\ b(r_i, g_i) &= a(g_i, f_i) = 1, & i \in S_c \\ b(r_i, g_i) &= a(g_i, f_i) - 1 + \beta = \beta, & i \in S_d \\ b(r_i, g_i) &\geq a(g_i, f_i) - 1 = 0, & i \in S_e \end{aligned} \tag{17}$$

where we have used Eqs. 12 and 15. Now

$$d_E(\vec{r},\vec{g}) = \sum_{i=1}^{n} b(r_i,g_i)$$

$$\geq \sum_{i\in S_o} a(g_i,f_i) + \sum_{i\in S_c} a(g_i,f_i) + \sum_{i\in S_d} [a(g_i,f_i)-1+\beta] + \sum_{i\in S_e} [a(g_i,f_i)-1]$$

$$= d_H(\vec{f},\vec{g}) - (1-\beta)|S_d| - |S_e|$$

$$\geq d - (1-\beta)s - t, \tag{18}$$

where we have used Eqs. 13, 16, 17 and the fact that the minimum Hamming distance between two code words is d. From Eqs. 14 and 18, we have proved that

$$d_E(\vec{r},\vec{g}) > d_E(\vec{r},\vec{f}) \text{ if } t + \beta s < d - (1-\beta)s - t \text{ or } 2t + s < d. \tag{19}$$

(The vanishing of β shows why we took it to be arbitrary.) Thus with a decoding rule based on Elias distance, we are assured of decoding correctly if $2t + s < d$, in perfect analogy to errors-only decoding. When we decode only out to the minimum distance – that is, when the distance criterion of (19) is apparently satisfied – we call this deletions-and-errors decoding.

That erasures could be used with minimum distance codes in this way has long been recognized, but few actual decoding schemes have been proposed. One of our chief concerns in Section III will be to develop a deletions-and-errors decoding algorithm for the important class of BCH codes. There we find that such an algorithm is very little more complicated than that appropriate to errors-only decoding.

2.3 GENERALIZED MINIMUM-DISTANCE DECODING

A further step in the same direction, not previously investigated, is to permit the inner decoder to classify its choice in one of a group of J reliability classes C_j, $1 \leq j \leq J$, rather than just two as previously. We define the generalized weight by

$$c(r_i,f_i) \equiv \begin{cases} \beta_{cj}, & r_i \text{ in class } C_j \text{ and } r_i = f_i \\ \beta_{ej}, & r_i \text{ in class } C_j \text{ and } r_i \neq f_i \end{cases} \tag{20}$$

where $0 \leq \beta_{cj} \leq \beta_{ej} \leq 1$. It will develop that only the difference

$$a_j \equiv \beta_{ej} - \beta_{cj}$$

of these weights is important; a_j will be called the reliability weight or simply weight corresponding to class C_j. We have $0 \leq a_j \leq 1$; a large weight corresponds to a class we consider quite reliable, and a small weight to a class considered unreliable; indeed,

if $\alpha_j < \alpha_k$ we shall say class C_j is less reliable than C_k. The case $\alpha_j = 0$ corresponds to an erasure, and of $\alpha_j = 1$ to the fully reliable symbols of the preceding section.

Let us now define a generalized distance

$$d_G(\vec{r},\vec{f}) \equiv \sum_{i=1}^{n} c(r_i,f_i). \tag{21}$$

Again we suppose the transmission of some word $\vec{f}$ from a code of minimum distance d, and the reception of a word in which n_{cj} symbols are received correctly and placed in class C_j, and n_{ej} are received incorrectly in C_j. Then

$$d_G(\vec{r},\vec{f}) = \sum_{j=1}^{J} [n_{ej}\beta_{ej}+n_{cj}\beta_{cj}]. \tag{22}$$

Take some other code word $\vec{g}$, and define the sets S_o, S_{cj}, and S_{ej} by

$$i \in \begin{cases} S_o & \text{if } f_i = g_i \\ S_{cj} & \text{if } f_i \neq g_i,\ r_i = f_i,\ r_i \text{ in class } C_j \\ S_{ej} & \text{if } f_i \neq g_i,\ r_i \neq f_i,\ r_i \text{ in class } C_j \end{cases} \tag{23}$$

Note that

$$\begin{aligned} |S_{cj}| &\leq n_{cj} \\ |S_{ej}| &\leq n_{ej}. \end{aligned} \tag{24}$$

Using Eqs. 20 and 23, we have

$$\begin{aligned} c(r_i,g_i) &\geq a(g_i,f_i) = 0, && i \in S_o \\ c(r_i,g_i) &= a(g_i,f_i) - 1 + \beta_{ej} = \beta_{ej}, && i \in S_{cj} \\ c(r_i,g_i) &\geq a(g_i,f_i) - 1 + \beta_{cj} = \beta_{cj}, && i \in S_{ej}, \end{aligned} \tag{25}$$

where the second relation depends on $r_i = f_i \neq g_i$, $i \in S_{cj}$. Now

$$d_G(\vec{r},\vec{g}) = \sum_{i=1}^{n} b(r_i,g_i)$$

$$\geq \sum_{i\in S_o} a(g_i,f_i) + \sum_{j=1}^{J} \left[\sum_{i\in S_{cj}} (a(g_i,f_i)-1+\beta_{ej}) + \sum_{i\in S_{ej}} (a(g_i,f_i)-1+\beta_{cj}) \right]$$

$$d_G(\vec{r},\vec{g}) = d_H(\vec{f},\vec{g}) - \sum_{j=1}^{J} [(1-\beta_{ej})|S_{cj}|+(1-\beta_{cj})|S_{ej}|]$$

$$\geq d - \sum_{j=1}^{J} [(1-\beta_{ej})n_{cj}+(1-\beta_{cj})n_{ej}]. \tag{26}$$

Thus, using Eqs. 22 and 26, we have proved that

$$d_G(\vec{r},\vec{g}) > d_G(\vec{r},\vec{f}) \quad \text{if} \quad \sum_{j=1}^{J} [(1-\beta_{ej}+\beta_{cj})n_{cj}+(1-\beta_{cj}+\beta_{ej})n_{ej}] < d,$$

$$\text{or} \quad \sum_{j=1}^{J} [(1-\alpha_j)n_{cj}+(1+\alpha_j)n_{ej}] < d. \tag{27}$$

Therefore if generalized distance is used as the decoding criterion, no decoding error will be made whenever n_{cj} and n_{ej} are such that the inequality of (27) is satisfied. When in addition we decode only out to the minimum distance – that is, whenever this inequality is apparently satisfied – we say we are doing generalized minimum-distance decoding.

This generalization is not interesting unless we can exhibit a reasonable decoding scheme that makes use of this distance criterion. The theorem that appears below shows that a decoder which can perform deletions-and-errors decoding can be adapted to perform generalized minimum-distance decoding.

We imagine that for the purpose of allowing a deletions-and-errors decoder to work on a received word, we make a temporary assignment of the weight $\alpha_j' = 1$ to the set of reliability classes C_j for which $j \in R$, say, and of the weight $\alpha_j' = 0$ to the remaining reliability classes C_j, $j \in E$, say. This means that provisionally all receptions in the classes C_j, $j \in E$, are considered to be erased, and all others to be reliable. We then let the deletions-and-errors decoder attempt to decode the resulting word, which it will be able to do if (see Eq. 27)

$$2 \sum_{j \in R} n_{ej} + \sum_{j \in E} (n_{cj}+n_{ej}) < d. \tag{28}$$

If it succeeds, it announces some code word which is within the minimum distance according to the Elias distance criterion of (28). We then take this announced word and see whether it also satisfies the generalized distance criterion of (27), now with the original weights α_j. If it does, then it is the unique code word within the minimum distance of the received word, and can therefore be announced as the choice of the outer decoder.

We are not guaranteed of succeeding with this method for any particular provisional assignment of the α_j'. The following theorem and its corollary show, however, that a

small number of such trials must succeed if the received word is within the minimum distance according to the criterion of Eq. 27.

Let the classes be ordered according to decreasing reliability, so that $\alpha_j \geq \alpha_k$ if $j < k$. Define the J-dimensional vector

$$\vec{\alpha} \equiv (\alpha_1, \alpha_2, \ldots, \alpha_J).$$

Let the sets R_a consist of all $j \leq a$, and E_a of all $j \geq a+1$, $0 \leq a \leq J$. Let $\vec{\alpha}'_a$ be the J-dimensional vector with ones in the first a places and zeros thereafter, which represents the provisional assignment of weights corresponding to $R = R_a$ and $E = E_a$. The idea of the following theorem is that $\vec{\alpha}$ is inside the convex hull whose extreme points are the $\vec{\alpha}'_a$, while the expression on the left in Eq. 27 is a linear function of $\vec{\alpha}$, which must take on its minimum value over the convex hull at some extreme point – that is, at one of the provisional assignments $\vec{\alpha}'_a$.

THEOREM: If $\sum_{j=1}^{J} \left[(1-\alpha_j)n_{cj}+(1+\alpha_j)n_{ej}\right] < d$ and $\alpha_j \geq \alpha_k$ for $j < k$, there is some integer a such that $2\sum_{j=1}^{a} n_{ej} + \sum_{j=a+1}^{J} (n_{cj}+n_{ej}) < d$.

Proof: Let

$$f(\vec{\alpha}) \equiv \sum_{j=1}^{J} \left[(1-\alpha_j)n_{cj}+(1+\alpha_j)n_{ej}\right].$$

Here, f is clearly a linear function of the J-dimensional vector $\vec{\alpha}$. Note that

$$f(\vec{\alpha}'_a) = 2\sum_{j=1}^{a} n_{ej} + \sum_{j=a+1}^{J} (n_{cj}+n_{ej}).$$

We prove the theorem by supposing that $f(\vec{\alpha}'_a) \geq d$, for all a such that $0 \leq a \leq J$, and exhibiting a contradiction. For, let

$$\lambda_0 \equiv 1 - \alpha_1$$

$$\lambda_a \equiv \alpha_a - \alpha_{a+1}, \quad 1 \leq a \leq J-1$$

$$\lambda_J \equiv \alpha_J.$$

We see that

$$0 \leq \lambda_a \leq 1, \; 0 \leq a \leq J, \text{ and } \sum_{a=0}^{J} \lambda_a = 1$$

so that the λ_a can be treated as probabilities. But now

$$\vec{\alpha} = \sum_{a=0}^{J} \lambda_a \vec{\alpha}'_a.$$

Therefore

$$f(\vec{\alpha}) = f\left(\sum_{a=0}^{J} \lambda_a \vec{\alpha}'_a\right) = \sum_{a=0}^{J} \lambda_a f(\vec{\alpha}'_a) \geq d \sum_{a=0}^{J} \lambda_a = d.$$

Thus if $f(\vec{\alpha}'_a) \geq d$, all a, then $f(\vec{\alpha}) \geq d$, in contradiction to the given conditions. Therefore $f(\vec{\alpha}'_a)$ must be less than d for at least one value of a. Q. E. D.

The import of the theorem is that if there is some code word which satisfies the generalized distance criterion of Eq. 27, then there must be some provisional assignment in which the least reliable classes are erased and the rest are not which will enable a deletions-and-errors decoder to succeed in finding that code word. But a deletions-and-errors decoder will succeed only if there are apparently no errors and $d - 1$ erasures, or one error and $d - 3$ erasures, and so forth up to t_o errors and $d - 2t_o - 1$ erasures, where t_o is the largest integer such that $2t_o \leq d - 1$. If by a trial we then mean an operation in which the $d - 1 - 2i$ least reliable symbols are erased, the resulting provisional word decoded by a deletions-and-errors decoder, and the resulting code word (if the decoder finds one) checked by Eq. 27, then we have the following corollary.

COROLLARY: $t_o + 1 \leq (d+1)/2$ trials suffice to decode any received word that is within the minimum distance by the generalized distance criterion of (27), regardless of how many reliability classes there are.

The maximum number of trials is then proportional only to d. Furthermore, many of the trials – perhaps all – may succeed, so that the average number of trials may be appreciably less than the maximum.

2.4 PERFORMANCE OF MINIMUM-DISTANCE DECODING SCHEMES

Our primary objective now is to develop exponentially tight bounds on the probability of error achievable with the three types of minimum-distance decoding discussed above, and with these bounds to compare the performance of the three schemes.

In the course of optimizing these bounds, however, we shall discover how best to assign the weights α_j to the different reliability classes. Since the complexity of the decoder is unaffected by the number of classes which we recognize, we shall let each distinguishable N-symbol sequence of outputs y_j form a separate reliability class, and let our analysis tell us how to group them. Under the assumption, as usual, that all code words are equally likely, the task of the inner decoder is to assign to the received y_j an x_j and an α_j, where x_j is the code word x for which $\Pr(y_j|x)$ is greatest, and α_j is the reliability weight that we shall determine.

a. The Chernoff Bound

We shall require a bound on the probability that a sum of independent, identically distributed random variables exceeds a certain quantity.

The bounding technique that we use is that of Chernoff[17]; the derivation which follows is due to Gallager.[18] This bound is known[19] to be exponentially tight, in the sense that no bound of the form $\Pr(e) \leq e^{-nE^*}$, where E^* is greater than the Chernoff bound exponent, can hold for arbitrarily large n.

Let y_i, $1 \leq i \leq n$, be n independent, identically distributed random variables, each with <u>moment-generating function</u>

$$g(s) \equiv \overline{e^{sy}} \equiv \sum \Pr(y)\, e^{sy},$$

and <u>semi-invariant moment-generating function</u>

$$\mu(s) \equiv \ln g(s).$$

Define y_{max} to be the largest value that y can assume, and

$$\overline{y} \equiv \sum y\Pr(y)$$

Let Y be the sum of the y_i, and let $\Pr(Y \geq n\delta)$ be the probability that Y exceeds $n\delta$, where $y_{max} \geq \delta \geq \overline{y}$. Then

$$\Pr(Y \geq n\delta) = \sum \Pr(y_1, y_2, \ldots, y_n)\, f(y_1, y_2, \ldots, y_n),$$

where

$$f(y_1, y_2, \ldots, y_n) = \begin{cases} 1, & Y = \sum_i y_i \geq n\delta \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, for any $s \geq 0$, we can bound f by

$$f(y_1, y_2, \ldots, y_n) \leq e^{s[Y-n\delta]}.$$

Then

$$\Pr(Y \geq n\delta) = \overline{f} \leq \overline{e^{sY}}\, e^{-ns\delta} = e^{-ns\delta}\, \overline{\prod_{i=1}^{n} e^{sy_i}}$$

$$= e^{-ns\delta} \prod_{i=1}^{n} \overline{e^{sy_i}}$$

$$= e^{-n[s\delta - \mu(s)]}, \qquad s \geq 0,$$

where we have used the fact that the average of a product of independent random variables is the product of their averages. To get the tightest bound, we maximize over s, and let

$$E(\delta) \equiv \max_{s \geq 0} \left[s\delta - \mu(s) \right].$$

Setting the derivative of the bracketed quantity to zero, we obtain

$$\delta = \mu'(s) = \frac{g'(s)}{g(s)}.$$

It can easily be shown that $\mu'(0) = \overline{y}$, $\mu'(\infty) = y_{max}$, and that $\mu'(s)$ is a monotonically increasing function of s. Therefore if $y_{max} \geq \delta \geq \overline{y}$, there is a non-negative s for which $\delta = \mu'(s)$, and substitution of this s in $(s\delta - \mu(s))$ gives $E(\delta)$.

As an example, which will be useful later, consider the variable y which takes on the value one with probability p and zero with probability $1 - p$. Then

$$g(s) = p\, e^s + 1 - p$$

$$\delta = \mu'(s) = \frac{p\, e^s}{p\, e^s + 1 - p}$$

$$e^s = \frac{\delta}{1-\delta} \, \frac{1-p}{p}$$

$$E(\delta) = \delta \ln \frac{\delta(1-p)}{p(1-\delta)} - \ln \frac{1-p}{1-\delta}$$

$$= -\delta \ln p - (1-\delta) \ln (1-p) - \mathcal{H}(\delta),$$

where

$$\mathcal{H}(\delta) \equiv -\delta \ln \delta - (1-\delta) \ln (1-\delta).$$

Then if $1 \geq \delta \geq p$,

$$\Pr(Y \geq n\delta) \leq e^{-n[-\delta \ln p - (1-\delta)\ln(1-p) - \mathcal{H}(\delta)]}.$$

This can be interpreted as a bound on the probability of getting more than $n\delta$ occurrences of a certain event in n independent trials, where the probability of that event in a single trial is p.

From this result we can derive one more fact which we shall need. Let $p = 1/2$, then

$$\Pr(Y \geq n\delta) = \sum_{i=n\delta}^{n} \binom{n}{i} 2^{-n} \leq 2^{-n} e^{n\mathcal{H}(\delta)}.$$

It follows that

$$\binom{n}{n\delta} \leq e^{n\mathcal{H}(\delta)}.$$

b. Optimization of Weights

We now show that the probability of decoding error or failure for minimum-distance decoding is the probability that a certain sum of independent identically distributed random variables exceeds a certain quantity, and therefore that we can use the Chernoff bound.

Let a code word from a code of length n and minimum distance d be transmitted. We know already that a minimum-distance decoder will fail to decode or decode incorrectly if and only if

$$\sum \left[n_{cj}(1-a_j)+n_{ej}(1+a_j)\right] \geq d \tag{29}$$

for, in the case of errors-only decoding, all $a_j = 1$; of deletions-and-errors decoding, $a_j = 0$ or 1; and of generalized minimum-distance decoding, $0 \leq a_j \leq 1$.

Under the assumption that the channel is memoryless and that there is no correlation between inputs, the probabilities p_{cj} of a correct reception in class C_j and p_{ej} of an incorrect reception in class C_j are constant and independent from symbol to symbol. Consider the random variable that for each symbol assumes the value $(1-a_j)$ if the symbol is received correctly and is given weight a_j, and $(1+a_j)$ if the symbol is received incorrectly and given weight a_j. These are then independent, identically distributed random variables with the common moment-generating function

$$g(s) = \sum \left[p_{cj}\, e^{s(1-a_j)} + p_{ej}\, e^{s(1+a_j)}\right]. \tag{30}$$

Furthermore, the condition of Eq. 29 is just the condition that the sum of these n random variables be greater than or equal to d. Letting $\delta = d/n$, we have by the Chernoff bound that the probability Pr(e) of error or failure is upperbounded by

$$\Pr(e) \leq e^{-nE'(\delta)}, \tag{31}$$

where

$$E'(\delta) \equiv \max_{s \geq 0} \left[s\delta - \mu(s)\right], \tag{32}$$

$\mu(s)$ being the natural logarithm of the g(s) of (30). This bound is valid for any particular assignment of the a_j to the reliability classes; however, we are free to vary the a_j to maximize this bound. Let

$$E(\delta) \equiv \max_{a_j} E'(\delta) = \max_{s, a_j} \left[s\delta - \mu(s)\right].$$

It is convenient and illuminating to maximize first over the distribution

$$E(\delta) = \max_{s} \left[s\delta - \mu_m(s)\right], \tag{33}$$

where

$$\mu_m(s) \equiv \min_{c_j} \mu(s) = \min_{c_j} \ln g(s) = \ln \min_{c_j} g(s) \equiv \ln g_m(s). \tag{34}$$

$\mu(s)$ is minimized by minimizing $g(s)$, and we shall now do this for the three types of minimum-distance decoding.

For errors-only decoding, there is no choice in the c_j, all of which must equal one; therefore,

$$g_m(s) = g(s) = e^{2s}\left[\sum_j p_{ej}\right] + \left[\sum_j p_{cj}\right]. \tag{35}$$

The total probability of symbol error is $p = \sum_j p_{ej}$. Making the substitutions $s' = 2s$ and $\delta' = \delta/2$, we see that this bound degenerates into the Chernoff bound of section 2.4a on getting more than $d/2$ symbol errors in a sequence of n transmissions, as might be expected.

For deletions-and-errors decoding, we can assign some outputs to a set E of erased symbols and the remainder to a set R of reliable symbols; we want to choose these sets so as to minimize $g(s)$. In symbols, $c_j = 0$, all $j \in E$, and $c_j = 1$, all $j \in R$, so

$$g(s) = e^{2s}\left[\sum_{j\in R} p_{ej}\right] + e^{s}\left[\sum_{j\in E} (p_{ej}+p_{cj})\right] + \left[\sum_{j\in R} p_{cj}\right].$$

Assigning a partic ar output y_j to E or R makes no difference if

$$e^{2s}\, p_{ej} + p_{cj} = e^{s}\,(p_{ej}+p_{cj})$$

or

$$L_j \equiv \frac{p_{ej}}{p_{cj}} = e^{-s}.$$

Here, we have defined L_j, the error-likelihood ratio, as p_{ej}/p_{cj}; we shall discuss the significance of L_j below. We see that to minimize $g(s)$, we let $j \in E$ if $L_j > e^{-s}$ and $j \in R$ if $L_j < e^{-s}$ — that is, comparison of L_j to a threshold that is a function of s is the optimum criterion of whether to erase or not. Then

$$g_m(s) = e^{2s}\, p_e(s) + e^{s}\, p_d(s) + p_e(s),$$

where

$$p_e(s) = \sum_{j \in R} p_{ej}; \qquad j \in R \text{ if } L_j \leq e^{-s}$$

$$p_d(s) = \sum_{j \in E} (p_{ej} + p_{cj}); \quad j \in E \text{ if } L_j > e^{-s} \tag{36}$$

$$p_c(s) = 1 - p_e(s) - p_d(s).$$

Finally, for generalized minimum-distance decoding, we have

$$g(s) = \sum_j \left[p_{cj} e^{s(1-\alpha_j)} + p_{ej} e^{s(1+\alpha_j)} \right],$$

which we can minimize with respect to a single α_j by setting the derivative

$$\frac{\partial g(s)}{\partial \alpha_j} = -sp_{cj} e^{s(1-\alpha_j)} + sp_{ej} e^{s(1+\alpha_j)}$$

to zero, as long as $0 \leq \alpha_j \leq 1$. The resulting condition is

$$e^{-2s\alpha_j} = \frac{p_{ej}}{p_{cj}} = L_j,$$

or

$$\alpha_j = -\frac{1}{2s} \ln L_j.$$

Whenever L_j is such that $-(\ln L_j)/2s > 1$, we let $\alpha_j = 1$, while whenever $-(\ln L_j)/2s < 0$, we let $\alpha_j = 0$. Then

$$g_m(s) = e^{2s} \left[\sum_{j \in R} p_{ej} \right] + \left[\sum_{j \in R} p_{cj} \right] + e^s \left[\sum_{j \in E} (p_{ej} + p_{cj}) \right] + e^s \left[\sum_{j \in G} 2\sqrt{p_{ej} p_{cj}} \right],$$

where

$$\begin{aligned} &j \in R \quad \text{if } L \leq e^{-2s}, \\ &j \in E \quad \text{if } L \geq 1, \\ &j \in G \quad \text{otherwise} \end{aligned} \tag{37}$$

and we have used $e^{s\alpha_j} = \sqrt{p_{cj}/p_{ej}}$ when $j \in G$.

Let us examine for a moment the error-likelihood ratio L_j. Denote by $\Pr(x_i, y_j)$ the probability of transmitting x_i and receiving y_j; the ratio L_{ij} between the probability that x_i was not transmitted, given the reception of y_j, and the probability that x_i <u>was</u> transmitted (the alternative hypothesis) is

$$L_{ij} = \frac{1 - \Pr(x_i|y_j)}{\Pr(x_i|y_j)} = \frac{\sum_{i'\neq i} \Pr(x_{i'}|y_j)}{\Pr(x_i|y_j)} = \frac{\sum_{i'\neq i} \Pr(x_{i'},y_j)}{\Pr(x_i,y_j)}.$$

The optimum decision rule for the inner decoder is to choose that x_i for which $\Pr(x_i|y_j)$ is maximum, or equivalently for which L_{ij} is minimum. But now for this x_i,

$$p_{cj} = \Pr(x_i,y_j) \text{ and } p_{ej} = \sum_{i'\neq i} \Pr(x_{i'},y_j).$$

Thus

$$L_j = \min_i L_{ij}.$$

We have seen that the optimum reliability weights are proportional to the L_j; thus the error-likelihood ratio is theoretically central to the inner decoder's decision making, both to its choice of a particular output and to its adding of reliability information to that choice. (The statistician will recognize the L_{ij} as sufficient statistics, and will appreciate that the simplification of minimum-distance decoding consists in its requiring of these statistics only the largest, and the corresponding value of i.)

The minimum value that L_j can assume is zero; the maximum, when all q inputs are equally likely, given y_j, is $q - 1$. When $q = 2$, therefore, L_j cannot exceed one. It follows that for generalized minimum-distance decoding with binary inputs the set E of Eq. 37 is empty.

In the discussion of the Chernoff bound we asserted that it was valid only when $\delta \geq \mu'(0)$, or in this case $\delta \geq \mu'_m(0)$. When $s = 0$, the sets R and E of (36) and (37) become identical, namely

$$j \in R' \quad \text{if } L \geq 1$$

$$j \in E \quad \text{if } L < 1.$$

Therefore $\mu'_m(0)$ is identical for deletions-and-errors and generalized minimum-distance decoding. If there is no output with $L_j < 1$ (as will always be true when there are only two inputs), then $\mu'_m(0)$ for these two schemes will equal that for errors-only decoding, too; otherwise it will be less. In the latter case, the use of deletions permits the probability of error to decrease exponentially with n for a smaller minimum distance $n\delta$, hence a larger rate, than without deletions.

We now maximize over s. From Eqs. 35-37, $\mu_m(s)$ has the general form

$$\mu_m(s) = \ln\left[e^{2s} p_2(s) + e^s p_1(s) + p_o(s)\right].$$

Setting the derivative of $(s\delta - \mu_m(s))$ to zero, we obtain

$$\delta = \mu'_m(s) = \frac{2e^{2s} p_2(s) + e^s p_1(s) + e^{2s} p'_2(s) + e^s p'_1(s) + p'_o(s)}{e^{2s} p_2(s) + e^s p_1(s) + p_o(s)} \tag{38}$$

which has a solution when $2 \geq \delta \geq \mu_m'(0)$. Substituting the value of s thus obtained in $(s\delta - \mu_m(s))$, we obtain $E(\delta)$, and thus a bound of the form

$$\Pr(\varepsilon) \leq e^{-nE(\delta)}. \tag{39}$$

We would prefer a bound that guaranteed the existence of a code of dimensionless rate r and length n with probability of decoding failure of error bounded by

$$\Pr(e) \leq e^{-nE(r)}.$$

The Gilbert bound[20] asserts for large n the existence of a code with a q-symbol alphabet, minimum distance δn, and dimensionless rate r, where

$$r \leq 1 - \frac{\mathcal{H}(\delta)}{\ln q} - \delta \frac{\ln (q-1)}{\ln q}.$$

Substitution of r for δ in (39) and using this relation with the equality sign gives us the bound we want.

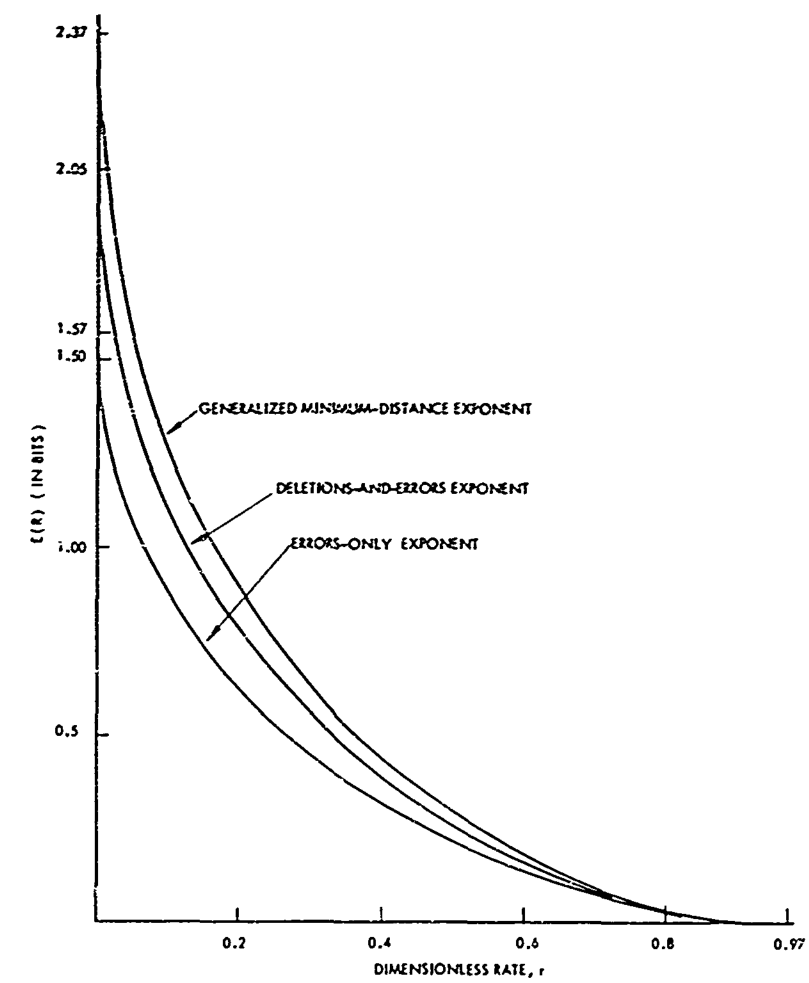

Fig. 3. Minimum-distance decoding exponents for a Gaussian channel with L = 3.

c. Computational Comparisons

To get some feeling for the relative performance of these three progressively more involved minimum-distance decoding schemes, the error exponents for each of them were computed over a few simple channels, with the use of the bounds discussed above.

In order to be able to compute easily the error-likelihood ratio, we considered only channels with two inputs. Figure 3 displays a typical result; these curves are for a channel with additive Gaussian noise of unit variance and a signal of amplitude either +3 or −3, which is a high signal-to-noise ratio. At lower signal-to-noise ratios the curves are closer. We also considered a two-dimensional Rayleigh fading channel for various signal-to-noise ratios.

For these channels, at least, we observed that though improvement is, of course, obtained in going from one decoding scheme to the next more complicated, this improvement is quite slight at high rates, and even at low rates, where improvement is greatest, the exponent for generalized minimum-distance decoding is never greater than twice that for errors-only decoding. The step between errors-only and deletions-and-errors decoding is comparable to, and slightly greater than, the step between the latter and generalized minimum-distance decoding.

From these computations and some of the computations that will be reported in Section VI, it would seem that the use of deletions offers substantial improvements in performance only when very poor outputs (with error-likelihood ratios greater than one) exist, and otherwise that only moderate returns are to be expected.

III. BOSE-CHAUDHURI-HOCQUENGHEM CODES

Our purpose now is to make the important class of BCH codes accessible to the reader with little previous background, and to do so with emphasis on the nonbinary BCH codes, particularly the RS codes, whose powerful properties are insufficiently known.

The presentation is quite single-minded in its omission of all but the essentials needed to understand BCH codes. The reader who is interested in a more rounded exposition is referred to the comprehensive and still timely book by Peterson.[4] In particular, our treatment of finite fields will be unsatisfactory to the reader who desires some depth of understanding about the properties that we assert; Albert[21] is a widely recommended mathematical text.

3.1 FINITE FIELDS

Mathematically, the finite field GF(q) consists of q elements that can be added, subtracted, multiplied, and divided almost as numbers. There is always a field element called zero (0), which has the property that any field element β plus or minus zero is β. There is also an element called one (1), such that $\beta \cdot 1 = \beta$; further, $\beta \cdot 0 = 0$. If β is not zero, it has a multiplicative inverse which is that unique field element satisfying the equation $\beta \cdot \beta^{-1} = 1$; division by β is accomplished by multiplication by β^{-1}.

The simplest examples of finite fields are the integers modulo a prime number p. For instance, take $p = 5$; there are 5 elements in the field, which we shall write I, II, III, IV, and V, to distinguish them from the integers to which they correspond. Addition, subtraction, and multiplication are carried out by converting these numbers into their integer equivalents and doing arithmetic modulo 5. For instance, I + III = IV, since $1 + 3 = 4 \bmod 5$; III + IV = II, since $3 + 4 = 2 \bmod 5$; I · III = III, since $1 \cdot 3 = 3 \bmod 5$; III · IV = II, since $3 \cdot 4 = 2 \bmod 5$. Figure 4 gives the complete addition and multiplication tables for GF(5).

+	I	II	III	IV	V
I	II	III	IV	V	I
II	III	IV	V	I	II
III	IV	V	I	II	III
IV	V	I	II	III	IV
V	I	II	III	IV	V

ADDITION TABLE

·	I	II	III	IV	V
I	I	II	III	IV	V
II	II	IV	I	III	V
III	III	I	IV	II	V
IV	IV	III	II	I	V
V	V	V	V	V	V

MULTIPLICATION TABLE

Fig. 4. Arithmetic in GF(5).

Note that V + β = β, if β is any member of the field; therefore, V must be the zero element. Also V · β = V. I · β = β, so I must be the one element. Since I · I = II · III = IV · IV = I, $\mathrm{I}^{-1} = \mathrm{I}$, $\mathrm{II}^{-1} = \mathrm{III}$, $\mathrm{III}^{-1} = \mathrm{II}$, and $\mathrm{IV}^{-1} = \mathrm{IV}$.

In Figure 5 by these rules we have constructed a chart of the first 5 powers of the field elements. Observe that in every case $\beta^5 = \beta$, while with the exception of the zero element V, $\beta^4 = \mathrm{I}$. Furthermore, both II and III have the property that their first four powers are distinct, and therefore yield the 4 nonzero field elements. Therefore if we let α denote the element II, say, $\mathrm{I} = \alpha^0 = \alpha^4$, $\mathrm{II} = \alpha$, $\mathrm{III} = \alpha^3$, and $\mathrm{IV} = \alpha^2$, which gives us a convenient representation of the field elements for multiplication and division, in the same way that the logarithmic relationship $x = 10^{\log_{10} x}$ gives us a convenient representation of the real numbers for multiplication and division.

β	β^2	β^3	β^4	β^5
I	I	I	I	I
II	IV	III	I	II
III	IV	II	I	III
IV	I	IV	I	IV
V	V	V	V	V

Fig. 5. Powers of the field elements.

	+, −	×, ÷
I	1	
II	2	
III	3	
IV	4	
V	0	0

Fig. 6. Representations for GF(5).

Figure 6 displays the two representations of GF(5) that are convenient for addition and multiplication. If β corresponds to a and α^b, and γ corresponds to c and α^d, then $\beta + \gamma \longleftrightarrow a + c \bmod 5$, $\beta - \gamma \longleftrightarrow a - c \bmod 5$, $\beta \cdot \gamma \longleftrightarrow \alpha^{[b+d \bmod 4]}$, and $\beta \cdot \gamma^{-1} \longleftrightarrow \alpha^{[b-d \bmod 4]}$, where $\longleftrightarrow$ means 'corresponds to' and the 'mod 4' in the exponent arises, since $\alpha^4 = \alpha^0 = 1$.

The prime field of most practical interest is GF(2), whose two elements are simply 0 and 1. Addition and multiplication tables for GF(2) appear in Fig. 7.

It can be shown[21] that the general finite field GF(q) has $q = p^m$ elements, where p is again a prime, called the characteristic of the field, and m is an arbitrary integer. As with GF(5), we find it possible to construct two representations of GF(q), one convenient for addition, one for multiplication. For addition, an element β of GF(q) is represented by a sequence of m integers, $b_1, b_2, \ldots, b_m$. To add β to α, we add b_1 to c_1, b_2 to c_2, and so forth, all modulo p. For multiplication, it is always possible to find a <u>primitive element</u> α, such that the first $q - 1$ powers of α yield the $q - 1$ nonzero field elements. Thus $\alpha^{q-1} = \alpha^0 = 1$ (or else the first $q - 1$ powers would not be distinct), and multiplication is accomplished by adding exponents mod $q - 1$. We have, if β is any nonzero element, $\beta^{q-1} \longleftrightarrow (\alpha^a)^{q-1} = (\alpha^{q-1})^a = 1^a = 1$, and thus for any β, zero or not, $\beta^q = \beta$.

	0	1
0	0	1
1	1	0

	0	1
0	0	0
1	0	1

Fig. 7. Tables for GF(2).

Thus all that remains to be specified is the properties of GF(q) to make the one-to-one identification between the addition and multiplication representations. Though this is easily done by using polynomials with coefficients from

GF(p),[4,21] it is not necessary to know precisely what this identification is to understand the sequel. (In fact, avoiding this point is the essential simplification of our presentation.) We note only that the zero element must be represented by a sequence of m zeros.

As an example of the general finite field, we use $GF(4) = GF(2^2)$, for which an addition table, multiplication table, and representation table are displayed in Fig. 8.

	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

ADDITION

	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	b	1
b	0	b	1	a

MULTIPLICATION

	+, -	×, ÷
0	00	0
1	01	
a	10	
b	11	

REPRESENTATIONS

Fig. 8. Tables for GF(4).

Note that GF(4) contains two elements that can be identified as the two elements of GF(2), namely 0 and 1. In this case GF(2) is said to be a subfield of GF(4). In general, GF((q')) is a subfield of GF(q) if and only if $q = q'^a$, where a is an integer. In particular, if $q = p^m$, the prime field GF(p) is a subfield of GF(q).

We shall need some explanation to understand our later comments on shortened RS codes. For addition, we have expressed the elements of GF(q) as a sequence of m elements from GF(p), and added place-by-place according to the addition rules of GF(p), that is, modulo p. Multiplication of an element of GF(q) by some member b of the subfield GF(p) amounts to multiplication by an integer b modulo p, which amounts to b-fold addition of the element of GF(q) to itself, which finally amounts to term-by-term multiplication of each of the m terms of the element by b mod p. (It follows that multiplication of any element of $GF(p^m)$ by p gives a sequence of zeros, that is, the zero element of $GF(p^m)$.) It is perhaps plausible that the following facts are true, as they are[21]: if $q = q'^a$, elements from GF(q) can always be expressed as a sequence of b elements from GF(q'), so that addition of 2 elements from GF(q) can be carried out place-by-place according to the rules of addition in GF(q'), and multiplication of an element from GF(q) by an element β from GF(q') can be carried out by term-by-term multiplication of each element in the sequence representing GF(q) by β according to the rules of multiplication in GF(q').

As an example, we can write the elements of GF(16) as

00	10	$\alpha 0$	$\alpha^2 0$
01	11	$\alpha 1$	$\alpha^2 1$
0α	1α	$\alpha\alpha$	$\alpha^2\alpha$
$0\alpha^2$	$1\alpha^2$	$\alpha\alpha^2$	$\alpha^2\alpha^2$

where α is a primitive element of GF(4). Then, by using Fig. 5, $(1\alpha) + (\alpha\alpha) = (\alpha^2 0)$, for example, while $\alpha \cdot (\alpha 1) = (\alpha^2 \alpha)$.

We have observed that $p \cdot \beta = 0$ for all elements in a field of characteristic p. In particular, if $p = 2$, $\beta + \beta = 0$, so that $\beta = -\beta$ and addition is the same as subtraction in a field characteristic two. Furthermore, $(\beta+\gamma)^p = \beta^p + \binom{p}{1} \beta^{p-1} + \dots + \binom{p}{p-1} \beta\gamma^{p-1} + \gamma^p$, by the binomial theorem; but every term but the first and last are multiplied by p, therefore zero, and $(\beta+\gamma)^p = \beta^p + \gamma^p$, when β and γ are elements of a field of characteristic p.

3.2 LINEAR CODES

We know from the coding theorem that codes containing an exponentially large number of code words are required to achieve an exponentially low probability of error. Linear codes[4, 22] can contain such a great number of words, yet remain feasible to generate; they can facilitate minimum distance decoding, as we shall see. Finally, as a class they can be shown to obey the coding theorem. They have therefore been overwhelmingly the codes most studied.

Assume that we have a channel with q inputs, where q is a prime power, so that we can identify the different inputs with the elements of a finite field GF(q). A <u>code word</u> $\vec{f}$ of length n for such a channel consists of a sequence of n elements from GF(q). We shall write $\vec{f} = (f_1, f_2, \dots, f_n)$, where f_i occupies the i^{th} <u>place</u>. The <u>weight</u> $w(\vec{f})$ of $\vec{f}$ is defined as the number of nonzero elements in $\vec{f}$.

A linear combination of two words $\vec{f}_1$ and $\vec{f}_2$ is written $\beta\vec{f}_1 + \gamma\vec{f}_2$, where β and γ are each elements of GF(q), and ordinary vectorial (that is, place-by-place) addition in GF(q) is implied. For example, if $\vec{f}_1 = (f_{11}, f_{12}, f_{13})$ and $\vec{f}_2 = (f_{21}, f_{22}, f_{23})$, then $\vec{f}_1 - \vec{f}_2 = (f_{11}-f_{21}, f_{12}-f_{22}, f_{13}-f_{23})$.

A <u>linear code</u> of length n is a subset of the q^n words of length n with the important property that any linear combination of words in the code yields another word in the code. A code is <u>nondegenerate</u> if all of its words are different; we consider only such codes.

Saying that the distance between two words $\vec{f}_1$ and $\vec{f}_2$ is d is equivalent to saying that the weight of their difference, $w(\vec{f}_1-\vec{f}_2)$, is d, since $\vec{f}_1 - \vec{f}_2$ will have zeros in places in which and only in which the two words do not differ. In a linear code, moreover, $\vec{f}_1 - \vec{f}_2$ must be another code word $\vec{f}_3$, so that if there are two code words separated by distance d there is a code word of weight d, and vice versa. Excluding the all-zero, zero-weight word, which must appear in every linear code, since $0 \cdot f_1 + 0 \cdot f_2$, is a valid linear combination of code words, and the minimum distance of a linear code is then the minimum weight of any of its words.

We shall be interested in the properties of sets of j different places, or sets of <u>size</u> j, which will be defined with reference to a given code. If the j places are such that there is no code word but the all-zero word with zeros in all j places, we say that these j places form a <u>non-null set</u> of size j for that code; otherwise they form a <u>null set</u>.

If there is a set of k places such that there is one and only one code word corresponding to each of the possible q^k assignments of elements from GF(q) to those k places,

then we call it an <u>information set</u>[23] of size k, thus any code with an information set of size k has exactly q^k code words. The remaining $n - k$ places form a <u>check set</u>. An information set must be a non-null set; for, otherwise there would be two or more words corresponding to the assignment of all zeros to the information set.

We now show that all linear codes have an information set, by showing the equivalence of the two statements: (i) there is an information set of size k for the code; (ii) the smallest non-null set has size k. For an information set of size k implies q^k code words; to any set of size $k - 1$ or less there are no more than q^{k-1} different assignments, and thus there must be at least two distinct code words that are the same in those places; but then their difference, though not the all-zero word, is zero in those places, so that any set of size $k - 1$ or less is a null set. Conversely, if the smallest non-null set has size k, then its every subset of $k - 1$ places is a null set; therefore there is a code word $\vec{f}$ that is zero in all but the p^{th} place, but is nonzero in the p^{th} place; if f has β in the p^{th} place, then $\beta^{-1} \cdot \vec{f}$ is a code word with a one in the p^{th} place, and zeros in the remaining information places. The k words with this property are called <u>generators</u>; clearly, their q^k linear combinations yield q^k code words that are distinct in the specified k places. (This is the property that makes linear codes easy to generate.) But there can be no more than q^k words in the code, otherwise all sets of size k would be null sets, by the arguments above. Thus the smallest non-null set must be an information set. Since every linear code has a smallest non-null set, every linear code has an information set and, for some k, q^k code words. In fact, every non-null set of size k is an information set, since to each of the q^k code words must correspond a different assignment of elements to those k places. We say such a code has k information symbols, $n - k$ check symbols, and dimensionless rate k/n, and call it an (n, k) code on GF(q).

If the minimum distance of a code is d, then the minimum weight of any non-zero code word is d, and the largest null set has size $n - d$. Therefore the smallest non-null set must have size $n - d + 1$ or less, so that the number of information symbols is $n - d + 1$ or less, and the number of check symbols $d - 1$ or greater. Clearly, we desire that for a given minimum distance k be as large as possible; a code that has length n, minimum distance d, and exactly the maximum number of information symbols, $n-d+1$, will be called a <u>maximum code</u>.[24]

We now show that a code is maximum if and only if every set of size $n - d + 1$ is an information set. For then no set of size $n - d + 1$ is a null set, thus no code word has weight $d - 1$ or less, and thus the minimum weight must be greater than or equal to d; but it cannot exceed d, since then there would have to be $n - d$ or fewer information symbols, so the minimum weight is d. Conversely, if the code is maximum, then the minimum weight of a code word is d, so that no set of size $n - d + 1$ can be a null set, but then all are information sets.

For example, let us investigate the code that consists of all words $\vec{f}$ satisfying the equation $f_1 + f_2 + \ldots + f_n = \sum_{i=1}^{n} f_i = 0$. It is a linear code, since if $\vec{f}_1$ and $\vec{f}_2$ satisfy this

equation, $\vec{f}_3 = (\beta\vec{f}_1+\gamma\vec{f}_2)$ also satisfies the equation. Let us assign elements from GF(q) arbitrarily to all places but the p^{th}. In order for there to be one and only one code word with these elements in these places, f_p must be the unique solution to

$$\sum_{i\neq p} f_i + f_p = 0, \quad \text{or } f_p = -\sum_{i\neq p} f_i.$$

Clearly, this specifies a unique f_p that solves the equation. Since p is arbitrary, every set of n − 1 places is thus an information set, so that this code is a maximum code with length n, n − 1 information symbols, and minimum distance 2.

a. Weight Distribution of Maximum Codes

In general, the number of code words of given weight in a linear code is difficult or impossible to determine; for many codes even d, the minimum weight, is not accurately known. Surprisingly, determination of the weight distribution of a maximum code presents no problems.

Suppose a maximum code of length n and minimum distance d, with symbols from GF(q); in such a code there are n − d + 1 information symbols, and, as we have seen, every set of n − d + 1 places must be an information set, which can be used to generate the complete set of code words.

Aside from the all-zero, zero-weight word, there are no code words of weight less than d. To find the number of code words of weight d, we reason as follows. Take an arbitrary set of d places, and consider the set of all code words that have all zeros in the remaining n − d places. One of these words will be the all-zero word; the rest must have weight d, since no code word has weight less than d. Consider the information set consisting of the n − d excluded places plus any place among the d chosen; by assigning zeros to the n − d excluded places and arbitrary elements to the last place we can generate the entire set of code words that have zeros in all n − d excluded places. There are thus q such code words, of which q − 1 have weight d. Since this argument obtains for an arbitrary set of d places, the total number of code words of weight d is $\binom{n}{d}(q-1)$.

Similarly, let us define by M_{d+a} the number of code words of weight d + a that are nonzero only in an arbitrary set of d + a places. Taking as an information set the n − d − a excluded places plus any a + 1 places of the d + a chosen, we can generate a total of q^{a+1} code words with all zeros in the n − d − a excluded places. Not all of these will have weight d + a, since for every subset of size d + i, $0 \leq i \leq a-1$, there will be M_{d+i} code words of weight d + i, all of which will have all zeros in the n − d − a excluded places. Subtracting also the all-zero word, we obtain

$$M_{d+a} = q^{a+1} - 1 - \sum_{i=0}^{a-1} \binom{d+a}{d+i} M_{d+i}.$$

From this recursion relation, there follows explicitly

$$M_{d+a} = (q-1) \sum_{i=0}^{a} (-1)^i \binom{d+a-1}{i} q^{a-i}.$$

Finally, since there are M_{d+a} words of weight $d + a$ in an arbitrary set of $d + a$ places, we obtain for N_{d+a}, the total number of code words of weight $d + a$,

$$N_{d+a} = \binom{n}{d+a} M_{d+a}.$$

We note that the summation in the expression for M_{d+a} is the first $a + 1$ terms of the binomial expansion of $(q-1)^{d+a-1} q^{-(d-1)}$, so that as $q \to \infty$, $M_{d+a} \to q^{a+1}$. Also, we may upperbound M_{d+a} by observing that when we generate the q^{a+1} code words that have all zeros in an arbitrary $n - d - a$ places, only those having no zeros in the remaining $a + 1$ information places have a chance of having weight $d + a$, so that

$$M_{d+a} \leq (q-1)^{a+1}.$$

3.3 REED-SOLOMON CODES

We can now introduce Reed-Solomon codes, whose properties follow directly from those of van der Monde matrices.

a. Van der Monde Matrices

An $(n+1) \times (n+1)$ van der Monde matrix has the general form:

$$\begin{bmatrix} 1 & a_o & a_o^2 & \cdots & a_o^n \\ 1 & a_1 & a_1^2 & \cdots & a_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^n \end{bmatrix}$$

where the a_i are members of some field. The determinant of this matrix, D, also a member of the field, is a polynomial in the a_i in which no a_i appears to a power greater than n. Furthermore, since the determinant is zero if any two rows are the same, this polynomial must contain as factors $a_i - a_j$, all $i \neq j$, so that

$$D = D' \prod_{i>j} (a_i - a_j).$$

But now the polynomial $\prod_{i>j} (a_i-a_j)$ contains each a_i to the n^{th} power, so that D' can only be a constant. Since the coefficient of $1 \cdot a_1 \cdot a_2^2 \cdots a_n^n$ in this polynomial must be one, $D' = 1$, and $D = \prod_{i>j} (a_i-a_j)$.

Now suppose that all the a_i are distinct. Then $a_i - a_j \neq 0$, $i \neq j$, since the a_i are members of a field. For the same reason, a product of nonzero terms cannot be zero, and therefore the determinant D is not zero if and only if the a_i are distinct.

Similarly,

$$\begin{vmatrix} a_0^{m_0} & a_0^{m_0+1} & \cdots & a_0^{m_0+n} \\ a_1^{m_0} & a_1^{m_0+1} & \cdots & a_1^{m_0+n} \\ \vdots & \vdots & & \vdots \\ a_n^{m_0} & a_n^{m_0+1} & \cdots & a_n^{m_0+n} \end{vmatrix} = \prod_i a_i^{m_0} \prod_{i>j} (a_i - a_j).$$

Thus the determinant of such a matrix, when $m_0 \neq 0$, is not zero if and only if the a_i are distinct and nonzero.

b. Reed-Solomon Codes

A Reed-Solomon[25] code on GF(q) consists of all words $\vec{f}$ of length $n \leq q - 1$ for which the $d - 1$ equations

$$\sum_{i=1}^{n} f_i \alpha^{im} = 0, \quad m_0 \leq m \leq m_0 + d - 2$$

are satisfied, where m_0 and d are arbitrary integers, and α is a primitive element of GF(q).

Clearly, an RS code is a linear code, since if $\vec{f}_1$ and $\vec{f}_2$ are code words satisfying the equations, $\beta\vec{f}_1 + \gamma\vec{f}_2 = f_3$ satisfies the equations. We shall now show that any $n - d + 1$ places of an RS code can be taken to be an information set, and therefore that an RS code is a maximum code with minimum distance d.

We define the <u>locator</u> Z_i of the i^{th} place as α^i; then we have $\sum_{i=1}^{n} f_i(Z_i)^m = 0$, $m_0 \leq m \leq m_0 + d - 2$. We note that since α is primitive and $n \leq q - 1$, the locators are distinct nonzero elements of GF(q). Let us arbitrarily assign elements of GF(q) to $n-d+1$ places; the claim is that no matter what the places, there is a unique code word with those elements in those places, and therefore any $n - d + 1$ places form an information set S. To prove this, we show that it is possible to solve uniquely for the symbols in the complementary check set $\overline{S}$, given the symbols in the information set. Let the locators of the check set $\overline{S}$ be Y_j, $1 \leq j \leq d - 1$, and the corresponding symbols be d_j. If there is a set of d_j that with the given information symbols forms a code word, then

$$\sum_{j=1}^{d-1} d_j(Y_j)^m = -\sum_{i \in S} f_i(Z_i)^m, \quad m_0 \leq m \leq m_0 + d - 2.$$

By defining $S_m \equiv -\sum_{i\in S} f_i(Z_i)^m$, these d − 1 equations can be written

$$\begin{bmatrix} Y_1^{m_o} & Y_1^{m_o+1} & \cdots & Y_1^{m_o+d-2} \\ Y_2^{m_o} & Y_2^{m_o+1} & \cdots & Y_2^{m_o+d-2} \\ \vdots & \vdots & & \vdots \\ Y_{d-1}^{m_o} & Y_{d-1}^{m_o+1} & \cdots & Y_{d-1}^{m_o+d-2} \end{bmatrix} \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_{d-1} \end{matrix} = \begin{bmatrix} S_{m_o} \\ S_{m_o+1} \\ \vdots \\ S_{m_o+d-2} \end{bmatrix}.$$

The coefficient matrix is of the van der Monde-like type that we examined above, and has nonzero determinant, since each of the locators is nonzero and distinct. Therefore there is a unique solution for the d_j for any assignment to the information places, so that an arbitrary set of n − d + 1 places can be taken as an information set. It follows that Reed-Solomon codes are maximum and have minimum distance d. The complete distribution of their weights has already been determined.

As examples, RS codes on GF(4) have length 3 (or less). The code of all words satisfying the single equation $f_1 + f_2 + f_3 = 0$ (m_o=0) has minimum distance 2. Taking the last symbol as the check symbol, we have $f_3 = f_1 + f_2$ (we omit minus signs, since we are in a field of characteristic two), so that the code words are

$$\begin{matrix} 000 & 101 & \alpha 0 \alpha & \alpha^2 0 \alpha^2 \\ 011 & 110 & \alpha 1 \alpha^2 & \alpha^2 1 \alpha \\ 0\alpha\alpha & 1\alpha\alpha^2 & \alpha\alpha 0 & \alpha^2 \alpha 1 \\ 0\alpha^2\alpha^2 & 1\alpha^2\alpha & \alpha\alpha^2 1 & \alpha^2\alpha^2 0 \end{matrix}$$

The code of all words satisfying $f_1 + f_2 + f_3 = 0$ and $f_1 + f_2\alpha + f_3\alpha^2 = 0$ (m_o=0) has minimum distance 3. Letting $f_2 = \alpha f_1$ and $f_3 = \alpha^2 f_1$, we get the code words

$$000 \quad 1\alpha\alpha^2 \quad \alpha\alpha^2 1 \quad \alpha^2 1 \alpha.$$

The code of all words satisfying $f_1 + f_2\alpha + f_3\alpha^2 = 0$ and $f_1 + f_2\alpha^2 + f_3\alpha^4 = 0$ (m_o=1) also has minimum distance 3; its code words are

$$000 \quad 111 \quad \alpha\alpha\alpha \quad \alpha^2\alpha^2\alpha^2.$$

c. Shortened RS Codes

A Reed-Solomon code can have length no longer than q − 1, for that is the total number of nonzero distinct elements from GF(q) which can be used as locators. (If m_o=0,

we can also let 0 be a locator, with the convention $0^0=1$, to get a code length q.) If we desire a code length $n \leq q-1$, we can clearly use any subset of the nonzero elements of GF(q) as locators.

Frequently, in concatenating codes, we meet the condition that q is very large, while n needs to be only moderately large. Under these conditions it is usually possible to find a subfield GF(q') of GF(q) such that $n < q'$. A considerable practical simplification then occurs when we choose the locators from the subfield of GF(q'). Recall that if $q'^b = q$, we can represent a particular symbol f_i by a sequence of b elements from GF(q'), $(f_{i1}, f_{i2}, \ldots, f_{ib})$. The conditions $\sum_i f_i Z_i^m = 0$, $m_o \leq m \leq m_o + d - 2$ then become the conditions $\sum_i f_{ij} Z_i^m = 0$, $m_o \leq m \leq m_o + d - 2$, $1 \leq j \leq b$, since when we add two f_i or multiply them by Z_i^m, we can do so term-by-term in GF(q'). In effect, we are interlacing b independent Reed-Solomon codes of length $n \leq q' - 1$. The practical advantage is that rather than having to decode an RS code defined on GF(q), we need merely decode RS codes defined on the much smaller field GF(q') b times. The performance of the codes cannot be decreased by this particular choice of locators, and may be improved if only a few constituent elements from GF(q') tend to be in error when there is an error in the complete symbol from GF(q).

As an example, if we choose $m_o = 1$ and use locators from GF(4) to get an RS code on GF(16) of length 3 and minimum distance 3, by using the representation of GF(16) in terms of GF(4), we get the 16 code words

$$\begin{pmatrix}000\\000\end{pmatrix}, \begin{pmatrix}000\\111\end{pmatrix}, \begin{pmatrix}000\\aaa\end{pmatrix}, \begin{pmatrix}0\ 0\ 0\\a^2a^2a^2\end{pmatrix}, \begin{pmatrix}111\\000\end{pmatrix}, \begin{pmatrix}111\\111\end{pmatrix}, \begin{pmatrix}111\\aaa\end{pmatrix}, \begin{pmatrix}1\ 1\ 1\\a^2a^2a^2\end{pmatrix},$$

$$\begin{pmatrix}aaa\\000\end{pmatrix}, \begin{pmatrix}aaa\\111\end{pmatrix}, \begin{pmatrix}aaa\\aaa\end{pmatrix}, \begin{pmatrix}a^2a^2a^2\\a^2a^2a^2\end{pmatrix}, \begin{pmatrix}a^2a^2a^2\\0\ 0\ 0\end{pmatrix}, \begin{pmatrix}a^2a^2a^2\\1\ 1\ 1\end{pmatrix}, \begin{pmatrix}a^2a^2a^2\\a\ a\ a\end{pmatrix}, \begin{pmatrix}a\ a\ a\\a^2a^2a^2\end{pmatrix},$$

or in effect two independent RS codes on GF(4).

3.4 BCH CODES

We shall now give a general method for finding a code with symbols from GF(q) of length n and minimum distance at least d. If $n \leq q-1$, of course, an RS code will be the best choice, since it is maximum. But often n is larger than q; for instance, if we want a binary code, $q = 2$, and the longest RS code has length one. BCH codes[26,27] are a satisfactory solution to this problem when n is not extravagantly large, and are the only general solution known.

Let us find an integer a such that $q^a > n$. Then there is an RS code on GF(q^a) with length n and minimum distance d. Since GF(q) is a subfield of GF(q^a), there will be a certain subset of the code words in this code with all symbols in GF(q). The minimum distance between any two words in this subset must be at least as great as the minimum distance of the code, so that this subset can be taken as a code on GF(q) with length n

and minimum distance at least d. Any such subset is a BCH code.

We shall call GF(q) the <u>symbol field</u> and $GF(q^a)$ the <u>locator field</u> of the code.

For example, from the three RS codes on GF(4) given as examples, we can derive the three binary codes:

a) 000	b) 000	c) 000
011		111
101		
110		

Since the sum of any two elements from GF(q) is another element in GF(q), the sum of any two words in the subset of code words with symbols from GF(q) is another word with symbols from GF(q), so that the subset forms a linear code. There must therefore be q^k words in the code, where k has yet to be determined. How useful the code is depends on how large k is; example b) shows that k can even be zero, and examples b) and c) show that k depends in general on the choice of m_o. We now show how to find the number of information symbols in a BCH code.

Since all code words are code words in the original RS code, all must satisfy the equations

$$\sum f_i Z_i^m = 0, \qquad m_o \leq m \leq m_o + d - 2.$$

Let the characteristic of the locator field $GF(q^a)$ be p; then $q^a = p^{am}$, $q = p^m$, and thus raising to the q^{th} power is a linear operation, $(\beta+\gamma)^q = \beta^q + \gamma^q$. Raising each side of these equations to the q^{th} power, we obtain

$$0 = \left(\sum_i f_i Z_i^m\right)^q = \sum_i f_i^q Z_i^{mq} = \sum_i f_i Z_i^{mq}, \qquad m_o \leq m \leq m_o + d - 2.$$

Here. we have used $f_i^q = f_i$ since f_i is an element of GF(q). Repeating this operation, we obtain

$$\sum_i f_i Z_i^{mq^j} = 0, \qquad 0 \leq j \leq a - 1, \tag{40}$$

where the process terminates at $j = a - 1$, since Z_i^m is an element of $GF(q^a)$, and therefore $\left(Z_i^m\right)^{q^a} = Z_i^m$. Not all of these equations are different, since if $mq^j = m'q^{j'}$ mod q − 1 for some $m' \neq m$, and $j' \neq j$, then $Z_i^{mq^j} = Z_i^{m'q^{j'}}$, for all i. Let us denote by r the number of equations that are distinct — that is, the number of distinct integers modulo q − 1 in the set

$$m_o,\ qm_o,\ q^2m_o,\ \ldots,\ q^{a-1}m_o$$

$$m_o + 1,\ q(m_o+1),\ \ldots,\ q^{a-1}(m_o+1)$$

$$m_o + d - 2,\ q(m_o+d-2),\ \ldots,\ q^{a-1}(m_o+d-2).$$

Clearly, $r \leq a(d-1)$. We label the distinct members of this set m_ℓ, $1 \leq \ell \leq r$.

We now show that r is the number of check symbols in the code. Let β be any element of $GF(q^a)$ with r distinct consecutive powers $\beta^b, \beta^{b+1}, \ldots, \beta^{b+r-1}$. We claim that the places whose locators are these r consecutive powers of β may be taken as a check set, and the remaining $n - r$ as an information set. Let the symbols in the information set S be chosen arbitrarily. A code word is uniquely determined by these information symbols if there is a unique solution to the r equations $\Sigma f_i(Z_i)^{m_\ell}$, $1 \leq \ell \leq r$, which in matrix form is

$$\begin{bmatrix} \beta^{bm_1} & \beta^{(b+1)m_1} & \cdots & \beta^{(b+r-1)m_1} \\ \beta^{bm_2} & \beta^{(b+1)m_2} & \cdots & \beta^{(b+r-1)m_2} \\ \vdots & \vdots & & \vdots \\ \beta^{bm_r} & \beta^{(b+1)m_r} & \cdots & \beta^{(b+r-1)m_r} \end{bmatrix} \begin{bmatrix} f_b \\ f_{b+1} \\ \vdots \\ f_{b+r-1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_r \end{bmatrix} \tag{41}$$

in which we have defined $S_\ell \equiv \sum_{i \in S} f_i Z_i^{m_\ell}$. The coefficient matrix is van der Monde-like (for a different reason than before), and since β^{m_ℓ} are all nonzero and distinct, the equations have the solution as claimed.

We must show that the f_{b+i} that solve Eqs. 41 are elements of the symbol field GF(q). Suppose we raise Eqs. 41 to the q^{th} power; we get a superficially new set of equations of the form

$$\sum f_i^q (Z_i)^{qm_\ell} = 0 \tag{42}$$

But for $i \in S$, $f_i \in GF(q)$, so $f_i^q = f_i$. Furthermore, Eqs. 42 are exactly the r distinct Eqs. 2, since Eqs. 2 are the distinct equations in Eqs. 1. Thus $f_b^q, f_{b+1}^q, \ldots, f_{b+r-1}^q$ solve Eqs. 41 for the same information symbols f_i, $i \in S$, as did $f_b, f_{b+1}, \ldots, f_{b+r-1}$, which were shown to be the unique solution to Eqs. 41. Therefore $f_{b+i}^q = f_{b+i}$; but the elements of $GF(q^a)$ which satisfy $\beta^q = \beta$ are precisely the elements of GF(q)[2], so that the f_{b+i} are elements of GF(q).

Thus the code has an information set of $n - r$ symbols, and therefore there are q^{n-r} code words.

We remark that any set of r places whose locators can be represented as r consecutive powers of some field element is thus a check set, and the remaining n − r an information set. In general every information set cannot be so specified, but this gives us a lower bound to their number.

For example, to find the number of check symbols in a binary code of length 15 ($q^a = 16$) and minimum distance 7, with m_o chosen as 1, we write the set

1, 2, 4, 8

3, 6, 12, 9 (24=9 mod 15)

5, 10 (20=5 mod 15)

where we have excluded all duplicates. There are thus 10 check symbols. This is the (15, 5) binary Bose-Chaudhuri[26] code.

a. Asymptotic Properties of BCH Codes

We recall that for large n the Gilbert bound guarantees the existence of a code with minimum distance n and dimensionless rate $k/n = 1 - \frac{\mathcal{H}(\delta)}{\ln q} - \delta \frac{\ln (q-1)}{\ln q}$. With a BCH code we are guaranteed to need no more than $a(d-1) = an\delta$ check symbols to get a minimum distance of at least $d = n\delta$, but since q^a must be greater than n, a must be greater than $\ln n/\ln q$, so that for any fixed nonzero δ, $an\delta$ exceeds n for very large n. Thus, at least to the accuracy of this bound, BCH codes are useless for very large n. It is well to point out, however, that cases are known in which the minimum distance of the BCH code is considerably larger than that of the RS code from which it was derived, and that it is suspected that their asymptotic performance is not nearly as bad as this result would indicate. But nothing bearing on this question has been proved.

IV. DECODING BCH CODES

We shall present here a decoding algorithm for BCH codes. Much of it is based on the error-correcting algorithm of Gorenstein and Zierler[26]; we have extended the algorithm to do deletions-and-errors and hence generalized minimum-distance decoding (cf. Section II). We have also appreciably simplified the final, erasure-correcting step.[27]

Since we intend to use a Reed-Solomon code as the outer code in all of our concatenation schemes, and minimization of decoder complexity is our purpose, we shall consider in Section VI in some detail the implementation of this algorithm in a special- or general-purpose computer.

Variations on this algorithm of lesser interest are reported in Appendix A.

4.1 INTRODUCTION

In Section III we observed that a BCH code is a subset of words from an RS code on GF(q) whose symbols are all members of some subfield of GF(q). Therefore we may use the same algorithm that decodes a certain RS code to decode all BCH codes derived from that code, with the proviso that if the algorithm comes up with a code word of the RS code which is not a code word in the BCH code being used, a decoding failure is detected.

Let us then consider the transmission of some code word $f = (f_1, f_2, \ldots, f_n)$ from a BCH code whose words satisfy

$$\sum_i f_i Z_i^m = 0, \qquad m_o \leq m \leq m_o + d - 2,$$

where the Z_i, the locators, are nonzero distinct elements of GF(q). In examples we shall use the RS code on GF(16) with $n = 15$, $m_o = 1$, and $d = 9$, and represent GF(16) as follows:

0	0000	α^3	0001	α^7	1101	α^{11}	0111
1	1000	α^4	1100	α^8	1010	α^{12}	1111
α	0100	α^5	0110	α^9	0101	α^{13}	1011
α^2	0010	α^6	0011	α^{10}	1110	α^{14}	1001

We shall let $Z_i = \alpha^{-i} = \alpha^{15-i}$.

We suppose that in the received word $\vec{r} = (r_1, r_2, \ldots, r_n)$, s symbols have been classed as unreliable, or erased. Let the locators of these symbols be Y_k, $1 \leq k \leq s$, and if the k^{th} deletion is in the i^{th} place, let $d_k = r_i - f_i$ be the value of the deletion, possibly zero. Also, of the symbols classed as reliable, let t actually be incorrect. Let the locators of these errors be X_j, $1 \leq j \leq t$, and if the j^{th} error is in the i^{th} place, let its value $e_j = r_i - f_i$, where now $e_j \neq 0$. We define the parity checks, or syndromes, S_m by

$$S_m \equiv \sum_i r_i Z_i^m;$$

then

$$S_m = \sum_i f_i Z_i^m + \sum_j e_j X_j^m + \sum_j d_k Y_k^m$$

$$= \sum_j e_j X_j^m + \sum_k d_k Y_k^m.$$

The decoding problem is to find the e_j, X_j, and d_k from the S_m and Y_k. The following algorithm solves this problem whenever $2t+s<d$.

We shall find it convenient in the sequel to define the column vectors

$$\vec{S}_{(a,b)} \equiv \left(S_a, S_{a-1}, \ldots, S_b\right)^T, \quad m_o \leq a \leq b \leq m_o + d - 2$$

$$\vec{X}_{j(a,b)} \equiv \left(X_j^a, X_j^{a-1}, \ldots, X_j^b\right)^T, \quad \text{and}$$

$$\vec{Y}_{k(a,b)} \equiv \left(Y_k^a, Y_k^{a-1}, \ldots, Y_k^b\right)^T.$$

Evidently,

$$\vec{S}_{(a,b)} = \sum_{j=1}^{t} e_j \vec{X}_{j(a,b)} + \sum_{k=1}^{s} d_k \vec{Y}_{k(a,b)}.$$

Finally, let us consider the polynomial $\sigma(Z) = 0$ defined by

$$\sigma(Z) \equiv (Z-Z_1)(Z-Z_2) \ldots (Z-Z_L),$$

where Z_ℓ are members of a field. Clearly $\sigma(Z) = 0$ if and only if Z equals one of the Z. Expanding (Z), we get

$$\sigma(Z) = Z^L - (Z_1+Z_2+\ldots+Z_L)Z^{L-1} + \ldots + (-1)^L (Z_1 Z_2 \ldots Z_L).$$

The coefficient of $(-1)^{L-\ell}Z^\ell$ in this expansion is defined as the $L-\ell^{th}$ elementary symmetric function $\sigma_{L-\ell}$ of $Z_1, Z_2, \ldots, Z_L$; note that σ_o is always one. We define $\vec{\sigma}$ as the row vector

$$\left(\sigma_o, -\sigma_1, \ldots, (-1)^L \sigma_L\right);$$

then the dot product

$$\vec{\sigma} \cdot \vec{Z}_{(L,o)} = \sigma(Z),$$

where

$$\vec{Z}_{(L,o)} \equiv \left(Z^L, Z^{L-1}, \dots, 1\right)^T.$$

4.2 MODIFIED CYCLIC PARITY CHECKS

The S_m are not the only parity checks that could be formed; in fact, any linear combination of the S_m is also a valid parity check. We look for a set of $d - s - 1$ independent parity-check equations which, unlike the S_m, do not depend on the erased symbols, yet retain the general properties of the S_m.

Let $\vec{\sigma}_d$ be the vector of the symmetric functions σ_{dk} of the erasure locators Y_k. We define the modified cyclic parity checks T_ℓ by

$$T_\ell \equiv \vec{\sigma}_d \cdot \vec{S}_{(m_o+\ell+s, m_o+\ell)}. \tag{40}$$

Since we must have $m \leq m_o + 1$ and $m_o + 1 + s \leq m_o + d - 2$, the range of 1 is $0 \leq 1 \leq d - s - 2$. In the case of no erasures, $T_\ell = S_{m_o+\ell}$. Now, since

$$\vec{S}_{(m_o+\ell+s, m_o+\ell)} = \sum_{j=1}^{t} e_j X_j^{m_o+\ell} \vec{X}_{j(s,o)} + \sum_{k=1}^{s} d_k Y_k^{m_o+\ell} \vec{Y}_{k(s,o)}, \tag{41}$$

we have

$$\begin{aligned} T_\ell \equiv \vec{\sigma}_d \cdot \vec{S}_{(m_o+\ell+s, m_o+\ell)} &= \sum_{j=1}^{t} e_j X_j^{m_o+\ell} \vec{\sigma}_d \cdot \vec{X}_{j(s,o)} + \sum_{k=1}^{s} d_k Y_k^{m_o+\ell} \vec{\sigma}_d \cdot \vec{Y}_{k(s,o)} \\ &= \sum_{j=1}^{t} e_j X_j^{m_o} \sigma_d(X_j) X_j^\ell + \sum_{k=1}^{s} d_k Y_k^{m_o+\ell} \sigma_d(Y_k) \\ &= \sum_{j=1}^{t} E_j X_j^\ell, \end{aligned} \tag{42}$$

in which we have defined $E_j \equiv e_j X_j^{m_o} \sigma_d(X_j)$ and used $\sigma_d(Y_k) = 0$, since Y_k is one of the erasure locators upon which $\vec{\sigma}_d$ is defined. The fact that the modified cyclic parity checks can be expressed as the simple function of the error locators given by Eq. 42 allows us to solve for the error locators in the same way as if there were no erasures and the minimum distance were $d - s$.

4.3 DETERMINING THE NUMBER OF ERRORS

If $d-s$ is odd, the maximum number of errors that can be corrected is $t_o = (d-s-1)/2$, while if $d-s$ is even, up to $t_o = (d-s-2)/2$ errors are correctable, and $t_o + 1$ are detectable.

We now show that the actual number of errors t is the rank of a certain $t_o \times t_o$ matrix M, whose components are modified cyclic parity checks, as long as $t \leq t_o$. In order to do this we use the theorem in algebra that the rank of a matrix is t if and only if there is at least one $t \times t$ submatrix with nonzero determinant, and all $(t+1) \times (t+1)$ submatrices have zero determinants. We also use the fact that the determinant of a matrix which is the product of square matrices is the product of the determinants of the square matrices.

THEOREM (after Gorenstein and Zierler[26]): If $t \leq t_o$, then M has rank t, where

$$M \equiv \begin{bmatrix} T_{2t_o-2} & T_{2t_o-3} & \cdots & T_{t_o-1} \\ T_{2t_o-3} & T_{2t_o-4} & \cdots & T_{t_o-2} \\ \vdots & \vdots & & \vdots \\ T_{t_o-1} & T_{t_o-2} & \cdots & T_o \end{bmatrix}.$$

Since $2t_o - 2 < d - s - 2$, all the T_ℓ in this matrix are available.

PROOF: First consider the $t \times t$ submatrix M_t formed by the first t rows and columns of M. Using Eq. 42, we can write M_t as the product of three $t \times t$ matrices as follows:

$$M_t = \begin{bmatrix} T_{2t_o-2} & T_{2t_o-3} & \cdots & T_{2t_o-t-1} \\ T_{2t_o-3} & T_{2t_o-4} & \cdots & T_{2t_o-3} \\ \vdots & \vdots & & \vdots \\ T_{2t_o-t-1} & T_{2t_o-t-2} & \cdots & T_{2t_o-2t} \end{bmatrix} = \begin{bmatrix} X_1^{t-1} & X_2^{t-1} & \cdots & X_t^{t-1} \\ X_1^{t-2} & X_2^{t-2} & \cdots & X_t^{t-2} \\ \vdots & \vdots & & \vdots \\ 1 & 1 & & 1 \end{bmatrix} \begin{bmatrix} E_1X_1^{2t_o-2t} & 0 & \cdots & 0 \\ 0 & E_2X_2^{2t_o-2t} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & E_tX_t^{2t_o-2t} \end{bmatrix} \begin{bmatrix} X_1^{t-1} & X_1^{t-2} & & 1 \\ X_2^{t-1} & X_2^{t-2} & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ X_t^{t-1} & X_t^{t-2} & & 1 \end{bmatrix}.$$

as may be checked by direct multiplication.

The center matrix is diagonal, and therefore has determinant $\prod_j E_jX_j^{2t_o-2t}$; since $E_j = e_jX_j^{m_o}\sigma_d(X_j)$, $X_j \neq Y_k$, and $e_j \neq 0$, this determinant is nonzero. The first and third matrices are van der Monde, with determinants $\prod_{i>j}(X_i - X_j)$, which is nonzero since the error locators are distinct. The determinant $|M_t|$ is then the product of three nonzero factors, and is therefore itself nonzero. Thus the rank of M is t or greater.

Now consider any of the $(t+1) \times (t+1)$ submatrices of M, which will have the general form

$$
\begin{bmatrix}
T_{a_0+b_0} & T_{a_0+b_1} & \cdots & T_{a_0+b_t} \\
T_{a_1+b_0} & T_{a_1+b_1} & \cdots & T_{a_1+b_t} \\
\vdots & \vdots & & \vdots \\
T_{a_t+b_0} & T_{a_t+b_1} & \cdots & T_{a_t+b_t}
\end{bmatrix}
=
\begin{bmatrix}
X_1^{a_0} & X_2^{a_0} & \cdots & X_t^{a_0} & 0 \\
X_1^{a_1} & X_2^{a_1} & \cdots & X_t^{a_1} & 0 \\
\vdots & \vdots & & \vdots & \vdots \\
X_1^{a_t} & X_2^{a_t} & \cdots & X_t^{a_t} & 0
\end{bmatrix}
\begin{bmatrix}
E_1 & 0 & \cdots & 0 & 0 \\
0 & E_2 & \cdots & 0 & 0 \\
\vdots & \vdots & & \vdots & \vdots \\
0 & 0 & \cdots & E_t & 0 \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
\begin{bmatrix}
X_1^{b_0} & X_1^{b_1} & \cdots & X_1^{b_t} \\
X_2^{b_0} & X_2^{b_1} & \cdots & X_2^{b_t} \\
\vdots & \vdots & & \vdots \\
X_t^{b_0} & X_t^{b_1} & \cdots & X_t^{b_t} \\
0 & 0 & \cdots & 0
\end{bmatrix}
$$

Again, this may be checked by direct multiplication with the use of Eq. 42. Each of the three factor matrices has an all-zero row and hence zero determinant; therefore all $(t+1) \times (t+1)$ submatrices of M have zero determinants. Thus the rank of M can be no greater than t; but then it is t.

4.4 LOCATING THE ERRORS

We now consider the vector $\vec{\sigma}_e$ of elementary symmetric functions σ_{e_j} of the X_j, and its associated polynomial

$$\sigma_e(x) = \vec{\sigma}_e \cdot \vec{X}_{(t,o)},$$

where

$$\vec{X}_{(t,o)} \equiv (X^t, X^{t-1}, \ldots, 1)^T .$$

If we could find the components of $\vec{\sigma}_e$, we could determine the error locators by finding the t distinct roots of $\sigma_e(X)$. If we define

$$\vec{T}_{(a,b)} \equiv \left(T_a, T_{a-1}, \ldots, T_b\right)^T, \qquad 0 \leqslant b \leqslant a \leqslant d-s-2,$$

then from Eq. 42

$$\vec{T}_{(a,b)} = \sum_{j=1}^{t} E_j X_{j(a,b)}$$

and we have

$$\vec{\sigma}_e \cdot \vec{T}_{(\ell'+t,\,\ell')} = \sum_{j=1}^{t} E_j X_j^{\ell'} \sigma_e(X_j) = 0, \qquad 0 \leqslant \ell' \leqslant d-s-t-2.$$

We know that the first component of $\vec{\sigma}_e$, σ_{e_o} equals one, so that this gives us a set of

$2t_o - t$ equations in t unknowns. Since $t \leq t_o$ by assumption, we can take the t equations specified by $2t_o - 2t \leq l' \leq 2t_o - t - 1$, which in matrix form are

$$-\begin{bmatrix} T_{2t_o-1} \\ T_{2t_o-2} \\ \vdots \\ T_{2t_o-t} \end{bmatrix} = \begin{bmatrix} T_{2t_o-2} & T_{2t_o-3} & \cdots & T_{2t_o-t-1} \\ T_{2t_o-3} & T_{2t_o-d} & \cdots & T_{2t_o-t-2} \\ \vdots & & & \vdots \\ T_{2t_o-t-1} & T_{2t_o-t-2} & \cdots & T_{2t_o-2t} \end{bmatrix} \begin{bmatrix} -\sigma_{e_1} \\ \sigma_{e_2} \\ \vdots \\ (-1)^t\sigma_{e_t} \end{bmatrix}$$

or, defining

$$\vec{\sigma}'_e = \left(-\sigma_{e_1}, \sigma_{e_2}, \ldots, (-1)^t\sigma_{e_t}\right),$$

$$-\vec{T}_{(2t_o-1, -t_o-t)} = \vec{\sigma}'_e M_t. \tag{43}$$

Since $0 \leq 2t_o - 2t$ and $2t_o - 1 \leq d - s - 2$, all of the T_ℓ needed to form these equations are available.

We have already shown that M_t has rank t, so that these equations are soluble for $\vec{\sigma}'_e$ and hence $\vec{\sigma}_e$. Then since $\sigma_e(Z_i)$ is zero if and only if Z_i is an error locator, calculation of $\sigma_e(Z_i)$ for each i will reveal in turn the positions of all t errors.

a. Remarks

The two steps of finding the rank of M and then solving a set of t equations in t equations in t unknowns may be combined into one. For, consider the equations

$$-\vec{T}_{(2t_o-1, t_o)} = \vec{\sigma}''_e M \tag{44}$$

where

$$\vec{\sigma}''_e \equiv \left(-\sigma_{e_1}, \sigma_{e_2}, \ldots, (-1)^t\sigma_{e_t}, 0, \ldots, 0\right).$$

An efficient way of solving (44) is by a Gauss-Jordan[28] reduction to upper triangular form. Since the rank of M is t, this will leave t nontrivial equations, the last $t_o - t$ equations being simply $0 = 0$. But now M_t is the upper left-hand corner of M, so that the upper left-hand corner of the reduced M will be the reduced M_t. Therefore, we can at this point set the last $t_o - t$ components of $\vec{\sigma}''_e$ to zero, and get a set of equations equivalent to (44), which can be solved for $\vec{\sigma}'_e$. Thus we need only one reduction, not two; since Gauss-Jordan recudtions tend to be tedious, this may be a significant saving.

This procedure works whenever $t \leq t_o$ — that is, whenever the received word lies

within distance t_o of some code word, not counting places in which there are erasures. It will generally be possible to receive words greater than distance t_o from any code word, and upon such words the procedure discussed above must fail. This failure, corresponding to a detectable error, will turn up either in the failure of Eq. 44 to be reducible to the form described above or in $\sigma_e(X)$, which has an insufficient number of nonzero roots.

Finally, if $d - s$ is even, the preceding algorithm will locate all errors when $t \leq t_o = (d-s-2)/2$. Also, if $t = t_o + 1$, an uncorrectable error can be detected by the nonvanishing of the determinant of the $t \times t$ matrix with T_{d-s-2} in the upper left, T_o in the lower right. Such an error would be detected anyway at some later stage in the correction process.

b. Example 1

Consider the (15,7) distance 9 RS code that has been introduced. Suppose there occur errors of value α^4 in the first position and α in the fourth position, and erasures of value 1 in the second position and α^7 in the third position.

$$\left(e_1 = \alpha^4, X_1 = \alpha^{14}, e_2 = \alpha, X_2 = \alpha^{11}, d_1 = 1, Y_1 = \alpha^{13}, d_2 = \alpha^7, Y_2 = \alpha^{12}\right).$$

In this case the parity checks S will turn out to be

$$S_1 = \alpha^{14},\ S_2 = \alpha^{13},\ S_3 = \alpha^5,\ S_4 = \alpha^6,\ S_5 = \alpha^9,\ S_6 = \alpha^{13},\ S_7 = \alpha^{10},\ \text{and } S_8 = \alpha^4.$$

With these eight parity checks and two erasure locators, the decoder must find the number and position of the errors. First it forms

$$\vec{\sigma}_d = \left(\sigma_{d_o}, \sigma_{d_1}, \sigma_{d_2}\right).$$

(Since we are working in a field of characteristic two, where addition and subtraction are identical, we omit minus signs.)

$$\sigma_{d_o} = 1$$

$$\sigma_{d_1} = Y_1 + Y_2 = \alpha^{13} + \alpha^{12} = (1011) + (1111) = (0100) = \alpha$$

$$\sigma_{d_2} = Y_1 Y_2 = \alpha^{13} \cdot \alpha^{12} = \alpha^{10}.$$

Next it forms the six modified cyclic parity checks T_ℓ by Eq. 41.

$$T_o = S_3 + \sigma_{d_1} S_2 + \sigma_{d_2} S_1 = \alpha^5 + \alpha \cdot \alpha^{13} + \alpha^{10} \cdot \alpha^{14} = \alpha^5 + \alpha^{14} + \alpha^9$$

$$= (0110) + (1001) + (0101) = (1010) = \alpha^8$$

$$T_1 = S_4 + \sigma_{d_1} S_3 + \sigma_{d_2} S_2 = \alpha^8$$

$$T_2 = 0, \quad T_3 = \alpha^3, \quad T_o = \alpha^{13}, \quad T_5 = \alpha^3.$$

Equations 44 now take the form

$$\begin{aligned}
\alpha^3 &= \alpha^{13}\sigma_{e_1} + \alpha^3\sigma_{e_2} \\
\alpha^{13} &= \alpha^3\sigma_{e_1} \qquad\qquad + \alpha^8\sigma_{e_3} \\
\alpha^3 &= \qquad\qquad \alpha^8\sigma_{e_2} + \alpha^8\sigma_{e_3}.
\end{aligned}$$

By reducing these equations to upper triangular form, the decoder gets

$$\begin{aligned}
\alpha^5 &= \sigma_{e_1} + \alpha^5\sigma_{e_2} \\
\alpha^{10} &= \qquad\quad \sigma_{e_2} + \sigma_{e_3} \\
0 &= 0.
\end{aligned}$$

From the vanishing of the third equation, it learns that only two errors actually occurred. Therefore it sets σ_{e_3} to zero and solves for σ_{e_1} and σ_{e_2}, obtaining

$$\sigma_{e_1} = \alpha^{10}, \quad \sigma_{e_2} = \alpha^{10}.$$

Finally, it evaluates the polynomial

$$\sigma_e(X) = X^2 + \sigma_{e_1} X + \sigma_{e_2} = X^2 + \alpha^{10} X + \alpha^{10},$$

for X equal to each of the nonzero elements of GF(16); $\sigma_e(X) = 0$ when $X = \alpha^{14}$ and $X = \alpha^{11}$, so that these are the two error locators.

4.5 SOLVING FOR THE VALUES OF THE ERASED SYMBOLS

Once the errors have been located, they can be treated as erasures. We are then interested in the problem of determining the values of s + t erased symbols, given that there are no errors in the remaining symbols. To simplify notation, we consider the problem of finding the d_k, given the Y_k, $1 \leq k \leq s$, and $t = 0$.

Since the parity-check equations are linear in the erasure values, we could solve s of them for the d. There is another approach, however, which is more efficient.

As an aid to understanding the derivation of the next equation, imagine the following situation. To find d_{k_o}, suppose we continued to treat the remaining s - 1 erasures as

erasures, but made a stab at guessing d_{k_o}. This would give us a word with s − 1 erasures and either one or (on the chance of a correct guess) zero errors. The rank of the matrix M_1 would therefore be either zero or one; but M_1 would be simply a single modified cyclic parity check, formed from the elementary symmetric functions of the s − 1 remaining erasure locators. Its vanishing would therefore tell us when we had guessed d_{k_o} correctly.

To derive an explicit formula, let ${}_{k_o}\vec{\sigma}_d$ be the vector of elementary symmetric functions of the s − 1 erasure locators, excluding Y_{k_o}. Since t = 0, we have from (41)

$$\vec{S}_{(m_o+d-2,\, m_o+d-s-1)} = \sum_{k=1}^{s} d_k Y_k^{m_o+d-s-1} \vec{Y}_{k(s-1,\,o)}$$

and therefore

$$\begin{aligned} {}_{k_o}T_{d-s-1} &\equiv {}_{k_o}\vec{\sigma}_d \cdot \vec{S}_{(m_o+d-2,\, m_o+d-s-1)} \\ &= d_{k_o} Y_{k_o}^{m_o+d-s-1}\, {}_{k_o}\sigma_d\!\left(Y_{k_o}\right) + \sum_{k\neq k_o} d_k Y_k^{m_o+d-s-1}\, {}_{k_o}\sigma_d(Y_k) \\ &= d_{k_o} Y_{k_o}^{m_o+d-s-1}\, {}_{k_o}\sigma_d\!\left(Y_{k_o}\right) \end{aligned}$$

since ${}_{k_o}\sigma_d(Y_k) = 0,\ k \neq k_o$. Thus

$$d_{k_o} = \frac{{}_{k_o}T_{d-s-1}}{Y_{k_o}^{m_o+d-s-1}\, {}_{k_o}\sigma_d\!\left(Y_{k_o}\right)}.$$

This gives us our explicit formula for d_{k_o}, valid for any s:

$$d_{k_o} = \frac{S_{m_o+d-2} - {}_{k_o}\sigma_{d_1} S_{m_o+d-3} + {}_{k_o}\sigma_{d_2} S_{m_o+d-4} - \cdots}{Y_{k_o}^{m_o+d-2} - {}_{k_o}\sigma_{d_1} Y_{k_o}^{m_o+d-3} + {}_{k_o}\sigma_{d_2} Y_{k_o}^{m_o+d-4} - \cdots}. \tag{45}$$

Evidently we can find all erasure values in this way; each requires the calculation of the symmetric functions of a different s − 1 locators. Alternatively, after finding d_1, we could modify all parity checks to account for this information as follows:

$$\left[\vec{S}'_{(m_o+d-2,\, m_o)} = \vec{S}_{(m_o+d-2,\, m_o)} - d_1 \vec{Y}_{1(m_o+d-2,\, m_o)}\right],$$

and solve for d_2 in terms of these new parity checks and the remaining $s-2$ erasure locators, and so forth.

A similar argument leads to the formula for error values,

$$e_{j_o} = \frac{{}_{j_o}\vec{\sigma}_e \cdot \vec{T}_{(d-s-2,\,d-s-t-1)}}{X_{j_o}^{m_o+d-s-t-1}\;{}_{j_o}\sigma_e\left(X_{j_o}\right)\sigma_d\left(X_{j_o}\right)},$$

in terms of the modified cyclic parity checks. We could therefore find all error values by this formula, modify the parity checks S_m accordingly, and then solve for the erasure values by Eq. 45.

a. Example 2

As a continuation of Example 1, let the decoder solve for e_1. The elementary symmetric functions of X_2, Y_1, and Y_2 are

$$\sigma_3 = X_2Y_1Y_2 = \alpha^6, \quad \sigma_2 = Y_2Y_1 + X_2Y_2 + X_2Y_1 = \alpha^3, \quad \sigma_1 + X_2 + Y_1 + Y_2 = \alpha^6.$$

Therefore

$$e_1 = \frac{\alpha^4 + \alpha^6 \cdot \alpha^{10} + \alpha^3 \cdot \alpha^{13} + \alpha^6 \cdot \alpha^9}{\alpha^7 + \alpha^6 \cdot \alpha^8 + \alpha^3 \cdot \alpha^9 + \alpha^6 \cdot \alpha^{16}} = \frac{\alpha}{\alpha^{12}} = \alpha^4.$$

e_2 can be found similarly, or the decoder can calculate

$$S_8' = S_8 + \alpha^4X_1^8 = \alpha^{13}, \quad S_7' = S_7 + \alpha^4X_1^7 = \alpha^3, \quad S_6' = S_6 + \alpha^4X_1^6 = 0.$$

Since

$$\sigma_2' = Y_1Y_2 = \alpha^{10}, \quad \sigma_1' = Y_1 + Y_2 = \alpha,$$

$$e_2 = \frac{\alpha^{13} + \alpha \cdot \alpha^3}{\alpha^{13} + \alpha \cdot \alpha^2 + \alpha^{10} \cdot \alpha^6} = \frac{\alpha^{11}}{\alpha^{10}} = \alpha.$$

Also, $S_8'' = \alpha^2$, $S_7'' = 0$,

so $d_1 = \dfrac{\alpha^2}{\alpha + \alpha^{12} \cdot \alpha^{13}} = 1,$

and, with $S_8''' = \alpha^{13}$, $d_2 = \dfrac{\alpha^{13}}{\alpha^6} = \alpha^7.$

4.6 IMPLEMENTATION

We now consider how a BCH decoder might be realized as a special-purpose computer. We shall assume the availability of an arithmetic unit able to realize, in

approximate order of complexity, the following functions of finite field elements: addition ($X=X_1+X_2$), squaring ($X=X^2$), multiplication by α^m, $m_o \leq m \leq m_o + d - 2$ ($X=\alpha^m X_1$). Furthermore, because of the bistability of common computer elements, we shall assume $p = 2$, so that subtraction is equivalent to addition and squaring is linear. We let the locators $Z_i = \alpha^{n-i}$. Finally, we shall assume that all elements of the symbol field are converted to their representations in the locator field $GF(q) = GF(2^M)$, and that all operations are carried out in the larger field.

Peterson[29] and Bartee and Schneider[30] have considered the implementation of such an arithmetic unit; they have shown that multiplication and inversion, the two most difficult operations, can be accomplished serially in a number of elementary operations proportional to M. All registers will be M bits long. Thus the hardware complexity is proportional to some small power of the logarithm of q, which exceeds the block length.

We attempt to estimate the approximate complexity of the algorithms described above by estimating the number of multiplications required by each and the number of memory registers.

During the computation, the received sequence of symbols must be stored in some buffer, awaiting correction. Once the S_m and Y_k have been determined, no further access to this sequence is required, until the sequence is read out and corrected.

The calculation of the parity checks

$$S_m \equiv r(\alpha^m) = r_1\alpha^{m(n-1)} + r_2\alpha^{m(n-2)} + \dots + r_n$$

is accomplished by the iteration

$$S_m = \left(\left(r_1\alpha^m + r_2\right)\alpha^m + r_3\right)\alpha^m + r_4 \cdots$$

which involves $n - 1$ multiplications by α^m. $d - 1$ such parity checks must be formed, requiring $d - 1$ registers.

$\vec{\sigma}_d$ can be calculated at the same time. We note that

$$\sigma_{d_k} = {}_{k_o}\sigma_{d_k} + Y_{k_o}\,{}_{k_o}\sigma_{d(k-1)};$$

$\vec{\sigma}_d$ can be calculated by this recursion relation as each new Y_k is determined. Adding a new Y_k requires s' multiplications when s' are already determined, so that the total number of multiplications, given s erasures, is

$$s - 1 + s - 2 + \dots = \binom{s}{2} < d^2/2.$$

s memory registers are required ($\sigma_{d_o} = 1$).

The modified cyclic parity checks T_ℓ are then calculated by Eqs. 40. Each requires s multiplications, and there are $d - s - 1$ of them, so that their calculation requires $s(d-s-1) < d^2/4$ multiplications and $d - s - 1$ memory registers.

Equations 44 are then set up in $t_o(t_o+1) < d^2/4$ memory registers. In the worst case,

$t = t_o$, the reduction to upper triangular form of these equations will require t_o inversions and

$$t_o(t_o+1) + (t_o-1)\, t_o + \dots + 1.2 = \frac{1}{4}\binom{2t_o+2}{3} + \binom{t_o+1}{2} < \frac{(t_o+1)^3}{3}$$

multiplications. As d becomes large, this step turns out to be the most lengthy, requiring as it does $\sim d^3/24$ multiplications.

Determination of $\vec{\sigma}_e$ from these reduced equations involves, in the worst case, a further $\binom{t_o}{2} < d^2/8$ multiplications, and t_o memory registers.

As Chien[31] has shown, finding the roots of $\sigma_e(X)$ is facilitated by use of the special multipliers by α^m in the arithmetic unit. If

$$\sum_{j=0}^{t} \sigma_e(t-j) = 0,$$

then 1 is a root of $\sigma_e(X)$. Let $\sigma'_{e(t-j)} = \alpha^{m_o+t-j}\,\sigma_{e(t-j)}$. Now

$$\sum_{j=0}^{t} \sigma'_{e(t-j)} = \alpha^{m_o+t} \sum_{j=0}^{t} \alpha^{-j}\sigma_{e(t-j)}$$

which will be zero when $\alpha^{-1} = \alpha^{n-1}$ is a root of $\sigma_e(X)$. All error locators can therefore be found with n multiplications by α^m, and stored in t memory registers.

Finally, we have only the problem of solving for s + t erasures. We use (45), which requires the elementary symmetric functions of all erasure locators but one. Since

$$_{k_o}\sigma_{d_k} = Y_{k_o}^{-1}\left(\sigma_{d(k+1)} - {}_{k_o}\sigma_{d(k+1)}\right),$$

we can begin with $_{k_o}\sigma_{d(s-1)} = Y_{k_o}^{-1}\sigma_{d_s}$ and find all $_{k_o}\sigma_{d_k}$ from the σ_{d_k} with s - 1 multiplications and an inversion. Then the calculation of (45) requires 2(s+t-1) multiplications and an inversion. Doing this s + t times, to find all erasure values, therefore requires 3(s+t)(s+t-1) multiplications and s + t inversions. Or we can alter s + t - 1 parity checks after finding the value of the first erasure, and repeat with s' = s + t - 1 and so forth; under the assumption that all $Y_{k_o}^m$ are readily available, this alternative requires only 2(s+t)(s+t-1) multiplications and s + t inversions.

a. Summary

To summarize, there are for any kind of decoding two steps in which the number of computations is proportional to n. If we restrict ourselves to correcting deletions only, then there is no step in which the number of computations is proportional to more than d^2. Otherwise, reduction of the matrix M requires some computations that may be as

large as d^3. If we are doing general minimum-distance decoding, then we may have to repeat the computation d/2 times, which leads to a total number of computations proportional to d^4. As for memory, we also have two kinds: a buffer with length proportional to n, and a number of live registers proportional to d^2. In sum, if $d = \delta n$, the total complixity of the decoder is proportional to n^b, where b is some number of the order of 3. All this suggests that if we are willing to use such a special-purpose computer as our decoder, or a specially programmed general-purpose machine, we can do quite powerful decoding without making the demands on this computer unreasonable.

Bartee and Schneider[32] built such a computer for a (127,92) 5-error-correcting binary BCH code, using the Peterson[33] algorithm. More recently, Zierler[34] has studied the implementation of his algorithm for the (255,225) 15-error-correcting Reed-Solomon code on GF(256), both in a special-purpose and in a specially programmed small general-purpose computer, with results that verify the feasibility of such decoders.

b. Modified Deletions-and-Errors Decoding

If a code has minimum distance d, up to $s_o = d - 1$ deletions may be corrected, or up to $t_o \leq (d-1)/2$ errors. We have seen that while the number of computations in the decoder was proportional to the cube of t_o, it is proportional only to the square of s_o. It may then be practical to make the probability of symbol error so much lower than that of symbol deletion that the probability of decoding error is negligibly affected when the decoder is set to correct only up to $t_1 < t_o$ errors. Such a tactic we call <u>modified deletions-and-errors decoding</u>, and we use it wherever we can in the computational program of Section VI.

V. EFFICIENCY AND COMPLEXITY

We shall now collect our major theoretical results on concatenated codes. We find that by concatenating we can achieve exponential decrease of probability of error with over-all block length, with only an algebraic increase in decoding complexity, for all rates below capacity; on an ideal superchannel with a great many inputs, Reed-Solomon codes can match the performance specified by the coding theorem; and with two stages of concatenation we can get a nonzero error exponent at all rates below capacity, although this exponent will be less than the unconcatenated exponent.

5.1 ASYMPTOTIC COMPLEXITY AND PERFORMANCE

We have previously pointed out that the main difficulty with the coding theorem is the complexity of the decoding schemes required to achieve the performance that it predicts. The coding theorem establishes precise bounds on the probability of error for block codes in terms of the length N of the code and its rate R. Informative as this theorem is, it is not precisely what an engineer would prefer, namely, the relationship between rate, probability of error, and complexity. Now complexity is a vague term, subsuming such incommensurable quantities as cost, reliability, and delay, and often depending on details of implementation. Therefore we should not expect to be able to discover more than rough relationships in this area. We shall investigate such relationships in the limit of very complex schemes and very low probabilities of error.

We are interested in schemes that have at least two adjustable parameters, the rate R and some characteristic length L, which for block codes will be proportional to the block length. We shall assume that the complexity of a scheme depends primarily on L. As L becomes large, a single term will always dominate the complexity. In the case in which the complexity is proportional to some algebraic function of L, or in which different parts of the complexity are proportional to algebraic functions of L, that part of the complexity which is proportional to the largest power of L, say L^{a}, will be the dominant contributor to the complexity when L is large, and we shall say the complexity is algebraic in L, or proportional to L. In the case in which some part of the complexity is proportional to the exponential of an algebraic function of L, this part becomes predominant when L is large (since $e^{x} = 1 + x + x^{2}/2! + \dots > x^{a}$, $x \to \infty$), and we say the complexity is exponential in L.

Similarly, the probability of error might be either algebraic or exponential in L, though normally it is exponentially small. Since what we are really interested in is the relationship between probability of error and complexity for a given rate, we can eliminate L from these two relationships in this way: if complexity is algebraic in L while Pr(e) is exponential in L, Pr(e) is exponential in complexity, while if both complexity and Pr(e) are exponential in L, Pr(e) is only algebraic in complexity.

For example, the coding theorem uses maximum-likelihood decoding of block codes of length N to achieve error probability $Pr(e) \le e^{-NE(R)}$. Maximum-likelihood decoding

involves e^{NR} comparisons, so that the complexity is also exponential in N. Therefore, Pr(e) is only algebraic in the complexity; in fact, if we let G be proportional to the complexity, $G = e^{NR}$, $(\ln G)/R = N$, $\Pr(e) \leq e^{-(\ln G)\frac{E(R)}{R}} = G^{-\frac{E(R)}{R}}$. As we have previously noted, this relatively weak dependence of Pr(e) on the complexity has retarded practical application of the coding theorem.

Sequential decoding of convolutional codes has attracted interest because it can be shown that for rates less than a critical rate $R_{comp} < C$, the average number of computations is bounded, while the probability of error approaches zero. The critical liability of this approach is that the number of computations needed to decode a given symbol is a random variable, and that therefore a buffer of length L must be provided to store incoming signals while the occasional long computation proceeds. Recent work[35] has shown that the probability of overflow of this buffer, for a given speed of computation, is proportional to $L^{-\alpha}$, where α is not large. In the absence of a feedback channel, buffer overflow is equivalent to system failure; thus the probability of such failure is only algebraically dependent upon the length of the buffer and hence on complexity.

Threshold decoding is another simple scheme for decoding short convolutional codes, but it has no asymptotic performance. As we have seen, BCH codes are subject to the same asymptotic deficiency. The only purely algebraic code discovered thus far that achieves arbitrarily low probability of error at a finite rate is Elias' scheme of iterating codes[36]; but this rate is low.

Ziv[37] has shown that by a three-stage concatenated code over a memoryless channel, a probability of error bounded by

$$\Pr(e) \leq K^{-L^{.5}}$$

can be achieved, where L is the total block length, while the number of computations required is proportional to L^{α}. His result holds for all rates less than the capacity of the original channel, although as $R \to C$, $\alpha \to \infty$.

In the sequel we shall show that by concatenating an arbitrarily large number of stages of RS codes with suitably chosen parameters on a memoryless channel, the overall probability of error can be bounded by

$$\Pr(e) \leq p_o^{L^{(1-\Delta)}},$$

where L is proportional to the total block length, and Δ is as small as desired, but positive. At the same time, if the complexity of the decoder for an RS code of length n is proportional to n^b, say, the complexity of the entire decoder is proportional to L^b. From the discussion in Section IV, we know that b is approximately 3. This result will obtain for all rates less than capacity.

We need a few lemmas to start. First, we observe that since a Reed-Solomon code of length n and dimensionless rate $(1-2\beta)$ can correct up to $n\beta$ errors, on a superchannel

with probability of error p,

$$\Pr(e) \leq \binom{n}{n\beta} p^{n\beta} \leq e^{-n[-\beta \log p - \mathcal{H}(\beta)]}. \tag{46}$$

Here, we have used a union bound and

$$\binom{n}{n\beta} \leq e^{n\mathcal{H}(\beta)}.$$

This is a very weak bound, but enough to show that the probability of error could be made to decrease exponentially with n for any β such that $-\beta \log p - \mathcal{H}(\beta) > 0$ if it were possible to construct an arbitrarily long Reed-Solomon code. In fact, however, if there are q inputs to the superchannel, with q a prime power, $n \leq q - 1$. We shall ignore the prime power requirement and the 'minus one' as trivial.

It is easily verified that for $\beta \leq 1/2$,

$$-\beta \ln \beta \geq -(1-\beta) \ln (1-\beta).$$

Therefore

$$-2\beta \ln \beta \geq \mathcal{H}(\beta) \geq -\beta \ln \beta, \qquad \beta \leq 1/2. \tag{47}$$

Now we can show that when $(-\ln \beta) \leq (2a)^{\frac{1}{a-1}}$,

$$\mathcal{H}(\beta^a) \leq \mathcal{H}^a(\beta) \tag{48}$$

For, by (47),

$$\mathcal{H}(\beta^a) \leq -2\beta^a \ln \beta^a = \beta^a \cdot 2a(-\ln \beta)$$

$$\mathcal{H}^a(\beta) \geq \beta^a(-\ln \beta)^a;$$

but

$$2ax \leq x^a \qquad \text{when } x \geq (2a)^{\frac{1}{a-1}}$$

which proves Eq. 48. We note that when $\beta \leq 1/e^2$, $a \geq 4$, this condition is always satisfied. (In fact, by changing the base of the logarithm, we can prove a similar lemma for any $\beta < 1$, $a > 1$.)

Finally, when $x > y > 0$, and $a > 1$,

$$(x-y)^a = x^a\left(1 - \left(\frac{y}{x}\right)^a\right) > x^a\left(1 - \frac{y}{x}\right) > x^a\left(1 - \frac{y}{x}\right)^a = x^a - y^a. \tag{49}$$

We are now ready to construct our many-stage concatenated code. Suppose by some block-coding scheme or otherwise we have achieved a superchannel with N_1 inputs and outputs and a probability of error

$$\Pr(e) \leq p_o \equiv e^{-E}, \quad E > 1 \tag{50}$$

We now apply to this superchannel an RS code of dimensionless rate $(1-2\beta)$ and length N_1, achieving a probability of error, from (46), of

$$\Pr_1(e) \leq e^{-N_1[\beta E - \mathcal{H}(\beta)]} \equiv e^{-E_1}. \tag{51}$$

Assume $\beta E - \mathcal{H}(\beta) > 0$, and define a to satisfy

$$N_1[\beta E - \mathcal{H}(\beta)] = E_1 \equiv E^a;$$

thus

$$a = \frac{\ln N_1}{\ln E} + \frac{\ln [\beta E - \mathcal{H}(\beta)]}{\ln E}. \tag{52}$$

We assume that

$$\beta \leq 1/e^2 \tag{53}$$

and

$$4 \leq a \leq N_1(1-2\beta),$$

and we shall prove the theorem only for these conditions.

This first concatenation creates a new superchannel with $N_1^{N_1(1-2\beta)}$ inputs and outputs and $\Pr(e) \leq \exp -E_1$. Apply a second RS code to this new superchannel of length $N_2 = N_1^a$ and dimensionless rate $(1-2\beta^a)$. (That a code of this length exists is guaranteed by the condition of Eq. 53 that $a \leq N_1(1-2\beta)$.) For this code,

$$\Pr(e) \leq e^{-N_2[\beta^a E_1 - \mathcal{H}(\beta^a)]} \equiv e^{-E_2}. \tag{54}$$

But now

$$\begin{aligned} E_2 = N_2[\beta^a E_1 - \mathcal{H}(\beta^a)] &= N_1^a[\beta^a E^a - \mathcal{H}(\beta^a)] \\ &\geq N_1^a[\beta^a E^a - \mathcal{H}^a(\beta)] \\ &\geq N_1^a[\beta E - \mathcal{H}(\beta)]^a \\ &= E_1^a. \end{aligned} \tag{55}$$

Here, we have used the inequalities of (48) and (49).

Thus by this second concatenation we achieve a code which, in terms of transmissions over the original superchannel, has length $N_1 N_2 = N_1^{a+1}$, dimensionless

rate $(1-2\beta)(1-2\beta^a)$, and $\Pr(e) \leq \exp -E^{a^2}$.

Obviously, if $\beta \leq 1/e^2$, then $\beta^a \leq 1/e^2$, and if $a \leq N_1(1-2\beta)$, then $a \leq N_2(1-2\beta^a)$. Therefore if we continue with any number of concatenations in this way, (53) remains satisfied, and relations like Eq. 55 obtain between any two successive exponents. After n such concatenations, we have a code of dimensionless rate $(1-2\beta)(1-2\beta^a)\ldots\left(1-2\beta^{a^{n-1}}\right)$ length $L = N_1^{\frac{a^{n-1}}{a-1}}$, and $\Pr(e) \leq \exp -E^{a^n}$. Now, for $a \geq 2$, $\beta < 1/2$,

$$
\begin{aligned}
(1-2\beta)(1-2\beta^a)\ldots\left(1-2\beta^{a^{n-1}}\right) &\geq (1-2\beta)(1-2\beta^2)\ldots\left(1-2\beta^{2^{n-1}}\right) \\
&= 1 - 2\beta - 2\beta^2 + 4\beta^3 - 2\beta^4 + \ldots \\
&\geq 1 - 2\beta - 4\beta^2 - 8\beta^3 - 16\beta^4 - \ldots \\
&= 1 - 2\beta\left(\frac{1}{1-2\beta}\right) = \frac{1-4\beta}{1-2\beta}.
\end{aligned} \tag{56}
$$

Also,

$$
\frac{a^{n-1}}{a-1}\ln N_1 = \ln L, \qquad a^n = 1 + (a-1)\frac{\ln L}{\ln N_1} \tag{57}
$$

so that

$$
\Pr(e) \leq e^{-E^{a^n}} = e^{-E\cdot E^{(a-1)\frac{\ln L}{\ln N_1}}} = p_o^{L^{(a-1)\frac{\ln E}{\ln N_1}}} = p_o^{L^{(1-\Delta)}}, \tag{58}
$$

by substitution for a, where Δ is defined by

$$
\Delta \equiv -\frac{\ln\left(\beta - \frac{\mathcal{H}(\beta)}{E}\right)}{\ln N_1}.
$$

Since $\beta E - \mathcal{H}(\beta)$ is assumed positive, but $\beta < 1$, Δ is positive.

We now construct a concatenated code of rate $R' \geq C(1-\epsilon)$ for a memoryless channel with error exponent $E(R)$. Choose $R = (1-\delta)C > R'$ and $\beta = \frac{\delta-\epsilon}{2(1+\delta-\epsilon)}$ so that $\frac{1-2\beta}{1-4\beta}R = C(1-\epsilon)$. We know there is some block code of length N and rate R such that $\Pr(e) \leq \exp -NE(R)$. Now we can apply the concatenation scheme already described with $N_1 = \exp NR$, $E = NE(R)$, as long as

$$
4 \leq a = \frac{NR}{\ln NE(R)} + \frac{\ln[\beta NE(R) - \mathcal{H}(\beta)]}{\ln NE(R)} \leq e^{NR}(1-2\beta).
$$

It is obvious that there is an N large enough so that this is true. Using this N, we

achieve a scheme with rate greater than or equal to $\frac{1-2\beta}{1-4\beta}R = C(1-\epsilon)$ and with probability of error

$$\Pr(e) \le e^{-NE(R)\cdot L^{(1-\Delta)}} = p_o^{L^{(1-\Delta)}},$$

$$\Delta = -\frac{\ln\left[\beta - \frac{\mathcal{H}(\beta)}{NE(R)}\right]}{NR}$$

Clearly, as long as $E(R) > 0$, Δ can be made as small as desired by letting N be sufficiently large. It remains positive, however, so that the error exponent E defined by

$$E \equiv \lim_{L\to\infty} -\frac{1}{L}\log \Pr(e)$$

appears to go to zero, if this bound is tight.

That E must be zero when an arbitrarily large number of minimum-distance codes are concatenated can be shown by the following simple lower bound. Suppose a code of length N can correct up to $N\beta$ errors; since the minimum distance cannot exceed N, $\beta \le 1/2$. Then on a channel with symbol probability of error p, a decoding error will certainly be made if the first $N\beta$ symbols are in error, so that

$$\Pr(e) \ge p^{N\beta}.$$

Concatenating a large number of such codes, we obtain

$$\Pr(e) \ge p_o^{(N_1N_2\dots)(\beta_1\beta_2\dots)}.$$

Now $N_1N_2\dots = L$, the total block length, so that

$$E \equiv \lim_{L\to\infty} -\frac{1}{L}\log \Pr(e) \le (-\log p_o) \lim (\beta_1\beta_2\dots) = 0,$$

because $\beta_i \le 1/2$. Since E cannot be less than zero, it must actually be zero. In other words, by concatenating an infinite number of RS codes, we can approach as close to a nonzero error exponent as we wish, for any rate less than capacity, but we can never actually get one.

As was shown in Section III, decoding up to t errors with an RS code requires a number of computations proportional to t^3. We require only that the complexity of a decoder which can correct up to $N\beta$ errors be algebraic in N, or proportional to N^b, although in fact it appears that $b \sim 3$. After going to n stages of concatenation according to the scheme above, the outermost decoder must correct $(N_1\beta)^{a^{n-1}}$ errors, the next outermost $(N_1\beta)^{a^{n-2}}$, and so forth. But in each complete block, the outermost decoder need

only compute once, while the next outermost decoder must compute $N_1^{a^{n-1}}$ times, the next outermost $N_1^{a^{n-1}} N_1^{a^{n-2}}$ times, and so forth. Hence the total number of computations is proportional to

$$G \sim \left[(N_1\beta)^{a^{n-1}}\right]^b + N_1^{a^{n-1}}\left[(N_1\beta)^{a^{n-2}}\right]^b + N_1^{a^{n-1}+a^{n-2}}\left[(N_1\beta)^{a^{n-3}}\right]^b + \dots$$

$$\leqslant N_1^{ba^{n-1}} + N_1^{a^{n-1}+ba^{n-2}} + N_1^{a^{n-1}+a^{n-2}+ba^{n-3}} + \dots .$$

Since $ba \geqslant b + a$, $a \geqslant 2$, $b \geqslant 2$, the first term in this series is dominant. Finally, since $N_1^{a^{n-1}} \leqslant L$,

$$G \lessapprox L^b.$$

Thus the number of computations can increase only as a small power of L. The complexity of the hardware required to implement these computations is also increasing, but generally only in proportion to a power of log L.

This result is not to be taken as a guide to design; in practice one finds it unnecessary to concatenate a large number of codes, as two stages generally suffice. It does indicate that concatenation is a powerful tool for getting exponentially small probabilities of error without an exponentially large decoder.

5.2 CODING THEOREM FOR IDEAL SUPERCHANNELS

We recall that an ideal superchannel is the q-input, q-output memoryless channel which is symmetric from the input and the output and has equiprobable errors. If its total probability of error is p, its transition probability matrix is

$$p_{ji} = \begin{cases} (1-p), & i = j \\ \dfrac{p}{q-1}, & i \neq j \end{cases} \tag{59}$$

We shall now calculate the unexpurgated part of the coding theorem bound for this channel, in the limit as q becomes very large. The result will tell us how well we can hope to do with any code when we assume we are dealing with an ideal superchannel. Then we shall find that over an interesting range Reed-Solomon codes are capable of achieving this standard. Finally, we shall use these results to compute performance bounds for concatenated codes.

Specialized to a symmetric discrete memoryless channel, the coding theorem asserts that there exists a code of length n and rate R which with maximum-likelihood decoding will yield a probability of error bounded by

$$Pr(e) \leqslant e^{-nE(R)},$$

where

$$E(R) = \max_{0<\rho\leq 1} \{E_o(\rho)-\rho R\} \tag{60}$$

and

$$E_o(\rho) = -\ln \sum_{j=1}^{q} \left[\sum_{i=1}^{q} \frac{1}{q} P_{ji}^{\frac{1}{1+\rho}} \right]^{1+\rho} . \tag{61}$$

Substituting Eq. 59 in Eq. 61, we obtain for the ideal superchannel

$$E_o(\rho) = -\ln q^{-\rho} \left[(1-p)^{\frac{1}{1+\rho}} + (q-1)^{\frac{\rho}{1+\rho}} p^{\frac{1}{1+\rho}} \right]^{1+\rho} \tag{62}$$

To facilitate handling Eq. 62 when q becomes large, we substitute $\rho' = \rho \ln q$ and the dimensionless rate $r = R/\ln q$; then

$$\Pr(e) \leq e^{-nE(r)}; \tag{63}$$

$$E(r) = \max_{0<\rho'\leq \ln q} \{E_o'(\rho')-\rho' r\}$$

$$E_o'(\rho') = -\ln e^{-\rho'} \left[(1-p)^{\frac{\ln q}{\ln q+\rho'}} + (q-1)^{\frac{\rho'}{\ln q+\rho'}} p^{\frac{\ln q}{\ln q+\rho'}} \right]^{1+\frac{\rho'}{\ln q}}$$

We consider first the case in which p is fixed, while q becomes very large. For $\rho' > 0$, $E_o'(\rho')$ becomes

$$E_o(\rho') = -\ln e^{-\rho'} [(1-p)+pe^{\rho'}]$$

$$= \rho' - \ln [(1-p)+pe^{\rho'}].$$

In the maximization of E(r), ρ' can now be as large as desired, so that the curved, unexpurgated part of the coding theorem bound is the entire bound; by setting the derivative of E(r) to zero, we obtain

$$r = \frac{\partial}{\partial \rho'} E_o(\rho')$$

$$= 1 - \frac{pe^{\rho'}}{(1-p) + pe^{\rho'}} = \frac{1-p}{(1-p) + pe^{\rho'}}$$

or

$$e^{\rho'} = \frac{1-p}{p} \cdot \frac{1-r}{r}.$$

Thus,

$$E(r) = (1-r)\ln\frac{1-p}{p}\cdot\frac{1-r}{r} - \ln\frac{1-p}{r}$$

$$= -r\ln(1-p) - (1-r)\ln p - \mathcal{H}(r). \tag{64}$$

This bound will be recognized as equal to the Chernoff bound – to the probability of getting greater than $n(1-r)$ errors in n transmissions, when the probability of error on any transmission is p. It suggests that a maximum-likelihood decoder for a good code corrects all patterns of $n(1-r)$ or fewer errors.

On the other hand, a code capable of correcting all patterns of $n(1-r)$ or fewer errors must have minimum distance $2n(1-r)$, thus at least $2n(1-r)$ check symbols, and dimensionless rate $r' = 1 - 2(1-r) < r$. No code of dimensionless rate r can correct all patterns of $n(1-r)$ or fewer errors. What must happen is that a good code corrects the great majority of error patterns beyond its minimum distance, out to $n(1-r)$ errors.

We shall show that on an ideal superchannel with q very large, Reed-Solomon codes do just about this, and come arbitrarily close to matching the performance of the coding theorem.

One way of approximating an ideal superchannel is to use a block code and decoder of length N and rate R over a raw channel with error exponent $E(R)$; then with e^{NR} inputs we have $\Pr(e) \leqslant e^{-NE(R)}$. We are thus interested in the case in which

$$q = e^{NR}$$

and (65)

$$p = e^{-NE}.$$

Substituting Eqs. 65 in Eqs. 63, and using $\rho' = \rho \ln q = \rho NR$, we obtain

$$\Pr(e) \leqslant e^{-nE(r)}$$

$$E(r) = \max_{0<\rho\leqslant 1}\{E_o(\rho) - \rho NRr\} \tag{66}$$

$$E_o(\rho) = -\ln e^{-\rho NR}\left[(1-e^{-NE})^{\frac{1}{1+\rho}} + (e^{NR}-1)^{\frac{\rho}{1+\rho}} e^{-\frac{NE}{1+\rho}}\right]^{1+\rho}.$$

When N becomes large, one or the other of the two terms within the brackets in this last equation dominates, and $E_o(\rho)$ becomes

$$E_o(\rho) = \begin{cases}\rho NR, & \rho NR \leqslant NE \\ NE, & NE \leqslant \rho NR,\end{cases}$$

or

$$E_o(\rho) = N\min\{\rho R, E\}. \tag{67}$$

The maximization of E(r) in (66) is achieved by setting $\rho = E/R$ if $E/R \leq 1$, and $\rho = 1$ otherwise. Thus

$$E(r) = \begin{cases} NE(1-r) & E \leq R \\ NR(1-r) & E \geq R \end{cases}$$

or

$$E(r) = N(1-r) \min\{E, R\}. \tag{68}$$

In the next section we shall only be interested in the case $E < R$, which corresponds to the curved portion of the bound, for which we have

$$\Pr(e) \leq e^{-nNE(1-r)}. \tag{69}$$

5.3 PERFORMANCE OF RS CODES ON THE IDEAL SUPERCHANNEL

We shall show that on an ideal superchannel (which suits RS codes perfectly), RS codes are capable of matching arbitrarily closely the coding theorem bounds, Eqs. 51 and 69, as long as q is sufficiently large. From these results we infer that RS codes are as good as any whenever we are content to treat the superchannel as ideal.

a. Maximum-Likelihood Decoding

We shall first investigate the performance of RS codes on a superchannel with large q and fixed p, for which we have shown (Eq. 51) that there exists a code with

$$\Pr(e) \leq e^{-n[-(1-r)\ln p - r\ln(1-p) - \mathcal{H}(r)]}.$$

It will be stated precisely in the following theorem.

THEOREM: For any $r > 1/2$, any δ such that $1/4 > \delta > 0$, and any p such that $1/4 > p > 0$, there exists a number Q such that for all ideal superchannels with probability of error p and $q \geq Q$ inputs, use of a Reed-Solomon code of length $n \leq q - 1$ and dimensionless rate r with maximum-likelihood decoding will result in a probability of error bounded by

$$\Pr(e) \leq 3e^{-n[-(1-r)\ln p - r\ln(1-p) - \mathcal{H}(r) - \delta]}.$$

PROOF: Let P_i be the probability that a decoding error is made, given i symbol errors. Then

$$\Pr(e) = \sum_{i=0}^{n} P_i \binom{n}{i} p^i (1-p)^{n-i}.$$

The idea of the proof is to find a bound for P_i which is less than one for $i \leq t$, and then to split this series into two parts,

$$\Pr(e) = \sum_{i=0}^{t} P_i \binom{n}{i} p^i (1-p)^{n-i} + \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}, \tag{70}$$

in which, because P_i falls off rapidly with decreasing i, the dominating term in the first series is the last, while that in the second series is the first.

We first bound P_i for $i \leq d-1$. Consider a single code word of weight w. By changing any k of its nonzero elements to zeros, any m of its nonzero elements to any of the other (q-2) nonzero field elements, and any l of its zero elements to any of the (q-1) nonzero field elements, we create a word of weight $i = w + l - k$, and at distance $j = k + l + m$ from the code word. The total number of words that can be so formed is

$$\binom{w}{k, m} \binom{n-w}{\ell} (q-2)^m (q-1)^{\ell}.$$

Here, the notation $\binom{w}{k, m}$ indicates the trinomial coefficient

$$\frac{w!}{k!\, m!\, (w-m-k)!}$$

which is the total number of ways a set containing w elements can be separated into subsets of k, m, and (w-m-k) elements. The total number, N, of words of weight i and distance j from some code word is then upperbounded by

$$N_{ij} \leq \sum_{\substack{w, k, \ell, m \\ i=w+\ell-k \\ j=k+\ell-m}} \binom{w}{k, m} \binom{n-w}{\ell} (q-2)^m (q-1)^{\ell} N_w, \tag{71}$$

where N_w is the total number of code words of weight w. The reason that this is an upper bound is that some words of weight i may be distance j from two or more code words.

We have shown (see Section III) that for a Reed-Solomon code,

$$N_w \leq \binom{n}{w} (q-1)^{w-d+1}.$$

Substituting this expression in (71) and letting $k = j - l - m$, $w = i + j - m - 2l$, we obtain

$$N_{ij} \leq \sum_{m \geq 0} \sum_{\ell \geq 0} \binom{i+j-m-2\ell}{j-\ell-m, m} \binom{n-i-j+m+2\ell}{\ell} \binom{n}{i+j-m-2\ell} (q-2)^m (q-1)^{i+j-m-\ell-d+1}$$

$$= \sum_{m \geq 0} \sum_{\ell \geq 0} \frac{n!\, (q-2)^m (q-1)^{i+j-m-\ell-d+1}}{m!\, \ell!\, (j-\ell-m)!\, (i-\ell-m)!\, (n-i-j+m+\ell)!}. \tag{72}$$

A more precise specification of the ranges of m and l is not necessary for our purposes.

The ratio of the $(l+1)^{\text{th}}$ to the l^{th} term in this series, for a given m,

$$\frac{(q-1)^{-1}\,(j-\ell-m)\,(i-\ell-m)}{(\ell+1)\,(n-i-j+m+\ell+1)}$$

is upperbounded by

$$\frac{(d-1)^2\,(q-1)^{-1}}{(\ell+1)\,[n-2(d-1)]} = \frac{n^2(1-r)^2}{(\ell+1)\,(q-1)\,n(2r-1)} \leqslant \frac{(1-r)^2}{(\ell+1)\,(2r-1)}.$$

Here, we have used $r > 1/2$, $j \leqslant i \leqslant d-1 = n(1-r)$, $l \geqslant 0$, $m \geqslant 0$, and $n \leqslant q-1$. Defining

$$C_1 \equiv \frac{(1-r)^2}{2r-1},$$

we have

$$N_{ij} \leqslant \sum_{m\geqslant 0} \frac{n!\,(q-2)^m\,(q-1)^{i+j-m-d+1}}{m!\,(j-m)!\,(i-m)!\,(n-i-j+m)!} \sum_{\ell\geqslant 0} \frac{C_1^{\ell}}{\ell!}$$

$$= e^{C_1} \sum_{m\geqslant 0} \frac{n!\,(q-2)^m\,(q-1)^{i+j-m-d+1}}{m!\,(j-m)!\,(i-m)!\,(n-i-j+m)!} \tag{73}$$

Similarly, the ratio of the $(m+1)^{\text{th}}$ to the m^{th} term in the series of (73),

$$\frac{(q-2)\,(j-m)\,(i-m)}{(q-1)\,(m+1)\,(n-i-j+m+1)},$$

is upperbounded by

$$\frac{(d-1)^2}{(m+1)\,[n-2(d-1)]} = \frac{nC_1}{(m+1)}$$

so that

$$N_{ij} \leqslant e^{C_1} \frac{n!\,(q-1)^{i+j-d+1}}{j!\,i!\,(n-i-j)!} \sum_{m\geqslant 0} \frac{(nC_1)^m}{m!}$$

$$= e^{C_1(n+1)} \binom{n}{i,j} (q-1)^{i+j-d+1}. \tag{74}$$

Since the total number of i-weight words is

$$\binom{n}{i}(q-1)^i,$$

the probability that a randomly chosen word of weight i will be distance j from some code word is bounded by

$$e^{C_1(n+1)}\binom{n-i}{j}(q-1)^{j+1-d},$$

and the total probability P_i that a word of weight i will be distance $j \leqslant i$ from some code word is bounded by

$$P_i \leqslant e^{C_1(n+1)} \sum_{j \leqslant i} \frac{(n-i)!\,(q-1)^{j+1-d}}{j!\,(n-i-j)!}$$

or, if we substitute $j' = i - j$,

$$P_i \leqslant e^{C_1(n+1)} \sum_{j' \geqslant 0} \frac{(n-i)!\,(q-1)^{i-j'+1-d}}{(i-j')!\,(n-2i+j')!}. \tag{75}$$

The ratio of the $(j'+1)^{\text{th}}$ to the j'^{th} term in the series of (75),

$$\frac{(i-j')}{(q-1)\,(n-2i+j'+1)},$$

is upperbounded by

$$C_2 = \frac{d-1}{(q-1)\,[n-2(d-1)]} = \frac{(1-r)}{(q-1)\,(2r-1)},$$

so that

$$P_i \leqslant e^{C_1(n+1)} \frac{(n-i)!\,(q-1)^{i+1-d}}{i!\,(n-2i)!} \sum_{j' \geqslant 0} C_2^{j'}.$$

If

$$q - 1 \geqslant 2\frac{(1-r)}{2r-1} \tag{76}$$

so that $C_2 \leqslant 1/2$,

$$\begin{aligned} P_i &\leqslant e^{C_1(n+1)}\binom{n-i}{i}(q-1)^{i+1-d} \\ &\leqslant 2e^{C_1(n+1)}\binom{n-i}{i}(q-1)^{i+1-d}. \end{aligned} \tag{77}$$

Substituting (77) in (70), we obtain

$$\Pr(e) \leq 2e^{C_1(n+1)} \sum_{i=0}^{t} \binom{n-i}{i} (q-1)^{i+1-d} \binom{n}{i} p^i(1-p)^{n-i} + \sum_{i=t+1}^{n} \binom{n}{i} p^i(1-p)^{n-i}$$

$$\equiv \qquad S_1 \qquad + \qquad S_2. \tag{78}$$

We let

$$\epsilon \equiv \frac{d-1-t}{n} > 0,$$

so that $t = n(1-r-\epsilon)$. The second series of (78) is just the probability that more than t errors occur, which is Chernoff-bounded by

$$S_2 \leq e^{-n[-(1-r-\epsilon)\ln p - (r+\epsilon)\ln(1-p) - \mathcal{H}(r+\epsilon)]}, \quad (1-r-\epsilon) > p. \tag{79}$$

(If $\epsilon < \delta$, $1 - r - \epsilon > 1/4 > p$.) Setting $i' = t - i$, we write the first series of (78) as

$$S_1 \equiv 2e^{C_1(n+1)} \sum_{i'\geq 0} \frac{n!\,(q-1)^{t+1-d-i'}\, p^{t-i'}(1-p)^{n-t+i'}}{(t-i')!\,(t-i')!\,(n-2t+2i')!}. \tag{80}$$

The ratio of the $(i'+1)^{th}$ to the i'^{th} term in the series of Eq. 80,

$$\frac{(1-p)\,(t-i')^2}{p(q-1)\,(n-2t+2i'+1)\,(n-2t+2i'+2)},$$

is upperbounded by

$$C_3 = \frac{(1-p)}{p(q-1)} \frac{(d-1)^2}{[n-2(d-1)]^2} = \frac{(1-p)(1-r)^2}{p(q-1)(2r-1)^2},$$

so that

$$S_1 \leq 2e^{C_1(n+1)} \frac{n!\,(q-1)^{t+1-d}\, p^t(1-p)^{n-t}}{t!\,t!\,(n-2t)!} \sum_{i'\geq 0} C_3^{i'}$$

if

$$q-1 \geq 2\,\frac{1-p}{p}\,\frac{(1-r)^2}{(2r-1)^2}, \tag{81}$$

so that $C_3 \leq 1/2$,

$$S_1 \leq 4e^{C_1(n+1)} \frac{n!\,(q-1)^{t+1-d}\, p^t(1-p)^{n-t}}{t!\,t!\,(n-2t)!}. \tag{82}$$

Substituting P_t from (77) in (82), we obtain

$$S_1 \leqslant 2P_t\binom{n}{t} p^t(1-p)^{n-t}. \tag{83}$$

Substituting (83) in (78), with the use of

$$\binom{n}{t} \leqslant e^{n\mathcal{H}(t/n)},$$

we have

$$\Pr(e) \leqslant (2P_t+1)\ e^{-n[-(1-r-\epsilon)\ln p-(r+\epsilon)\ln(1-p)-\mathcal{H}(r+\epsilon)]}.$$

Choose

$$\epsilon = \frac{\delta}{\ln(1-p) - \ln p} \tag{84}$$

since $p < 1/4$, $\epsilon < \delta$, and Eq. 79 is valid. Since

$$\mathcal{H}(r+\epsilon) \leqslant \mathcal{H}(r), \qquad r \geqslant 1/2,$$

$$\Pr(e) \leqslant (2P_t+1)\ e^{-n[-(1-r)\ln p-r\ln(1-p)-\mathcal{H}(r)]}. \tag{85}$$

Finally, for this choice of ϵ, from (77),

$$P_t = 2e^{C_1(n+1)} \binom{n-t}{t} (q-1)^{t+1-d}$$

$$\leqslant e^{n[C_1+1-\epsilon\ln(q-1)]+[C_1+\ln 2]}$$

in which we have used $d - 1 - t = n\epsilon$ and

$$\binom{n-t}{t} \leqslant e^{n\mathcal{H}\left(\frac{n-t}{t}\right)} \leqslant e^n.$$

Thus $P_t \leqslant 1$ if

$$\ln(q-1) \geqslant \frac{1}{\epsilon}\left[C_1 + 1 + \frac{C_1 + \ln 2}{n}\right]$$

$$\geqslant \frac{\ln(1-p) - \ln p}{\delta}\ [2C_1 + 1 + \ln 2], \tag{86}$$

in which we have used $n \geqslant 1$ and substituted for ϵ by Eq. 84. Defining $C_4 \equiv 2C_1 + 1 - \ln 2$, (84) can be written

$$q - 1 \geqslant \left[\frac{1-p}{p}\right]^{C_4/\delta}. \tag{87}$$

When this is satisfied,

$$\Pr(e) \leqslant 3e^{-n[-(1-r)\ln p-r\ln(1-p)-\mathcal{H}(r)-\delta]} \tag{88}$$

as was to be proved. Equation 88 holds if (76), (81), and (87) are simultaneously satisfied, which is to say if $q - 1 > Q$, with

$$Q \equiv \max\left\{2\frac{(1-r)}{2r-1},\ 2\frac{1-p}{p}\frac{(1-r)^2}{(2r-1)^2},\ \left[\frac{1-p}{p}\right]^{C_4/\delta}\right\}. \tag{89}$$

Q. E. D.

From this result we can derive a corollary that applies to the case in which $q = e^{NR}$, $p = e^{-NE}$, for which we have found the coding theorem bound, when $E < R$ (Eq. 69),

$$\Pr(\epsilon) \leq e^{-nNE(1-r)}.$$

COROLLARY: For $E < R$, $r > 1/2$, and any $\delta' > 0$, there exists an N_o such that for all $N \geq N_o$, use of a Reed-Solomon code of dimensionless rate r and length $n \leq q - 1$ with maximum-likelihood decoding on an ideal superchannel with probability of error $p = e^{-NE}$ and $q = e^{NR}$ inputs will yield an over-all probability of error bounded by

$$\Pr(\epsilon) \leq 3e^{-nN[E(1-r)-\delta']}.$$

Proof: The proof follows immediately from the previous theorem if we let $\delta = N\delta' - \mathcal{H}(r)$, which will be positive for

$$N > \frac{\mathcal{H}(r)}{\delta'}. \tag{90}$$

For then, since $-r \ln(1-p) \geq 0$,

$$\Pr(e) \leq 3e^{-n(1-r)NE+nN\delta'} \tag{91}$$

which was to be proved. Equation 91 holds if Eq. 90 holds and if, by substituting in Eq. 89,

$$e^{NR} \geq \max\left\{2\frac{1-r}{2r-1},\ 2\frac{1-e^{-NE}}{e^{-NE}}\frac{(1-r)^2}{(2r-1)^2},\ \left[\frac{1-e^{-NR}}{e^{-NE}}\right]^{\frac{C_4}{N\delta'\mathcal{H}(r)}}\right\}. \tag{92}$$

The first condition of (92) is satisfied if

$$N \geq \frac{1}{R}\ln 2\frac{1-r}{2r-1}; \tag{93}$$

the second, if

$$NR \geq \left[NE + \ln 2\frac{(1-r)^2}{(2r-1)^2}\right] \tag{94}$$

in which we have used $1 - e^{-NE} \leq 1$. Equation 94 can be rewritten

$$N \geq \frac{\ln 2 \dfrac{(1-r)^2}{(2r-1)^2}}{R - E}. \tag{95}$$

Here, we assume $R > E$.

The third condition of (92) is satisfied if

$$NR \geq NE\left[\frac{C_4}{N\delta' - \mathcal{H}(r)}\right], \tag{96}$$

which can be rewritten

$$N \geq \frac{EC_4/R + \mathcal{H}(r)}{\delta'}. \tag{97}$$

Equations 90, 93, 95, and 97 will be simultaneously satisfied if $N \geq N_o$, where

$$N_o \equiv \max\left\{\frac{\mathcal{H}(r)}{\delta'}, \frac{1}{R}\ln 2\frac{(1-r)}{2r-1}, \frac{1}{R-E}\ln 2\frac{(1-r)^2}{(2r-1)^2}, \frac{EC_4 + R\mathcal{H}(r)}{R\delta'}\right\}.$$

Q.E.D.

This result then provides something for communication theory which was lacking previously: a limited variety of combinations of very long codes and channels which approximate the performance promised by the coding theorem.

For our present interest, this result tells us that once we have decided to concatenate and to treat errors in the superchannel as equiprobable, a Reed-Solomon code is entirely satisfactory as an outer code. If we fail to meet coding-theorem standards of performance, it is because we choose to use minimum-distance rather than maximum-likelihood decoding.

b. Minimum-Distance Decoding

If we use minimum-distance decoding, decoding errors occur when there are $d/2 = n(1-r)/2$ or more symbol errors, so by the Chernoff bound

$$\Pr(e) \leq e^{-n\left[-\left(\frac{1-r}{2}\right)\ln p - \left(\frac{1+r}{2}\right)\ln(1-p) - \mathcal{H}\left(\frac{1-r}{2}\right)\right]}. \tag{98}$$

One way of interpreting this is that we need twice as much redundancy for minimum-distance decoding as for maximum-likelihood decoding. Or, for a particular dimensionless rate r, we suffer a loss of a factor K in the error exponent, where K goes to 2 when p is very small, and is greater than 2 otherwise. Indeed, when $q = e^{NR}$, $p = e^{-NE}$, and $E < R$, the loss in the exponent is exactly a factor of two, for (98) becomes

$$\Pr(e) \leq e^{-nNE(1-r)/2}.$$

5.4 EFFICIENCY OF TWO-STAGE CONCATENATION

By the coding theorem, we know that for any memoryless channel there is a code of length N' and rate R' such that $\Pr(e) \le e^{-N'E(R')}$, where $E(R')$ is the error exponent of the channel. We shall now show that over this same channel there exists an inner code of length N and rate R and an outer code of length n and dimensionless rate r, with $nN = N'$ and $rR = R'$, which when concatenated yield $\Pr(e) \le e^{-N'E_C(R')}$. We define the <u>efficiency</u> $\eta(R') \equiv E_C(R')/E(R')$; then, to the accuracy of the bound, the reciprocal of the efficiency indicates how much greater the over-all length of the concatenated code must be than that of a single code to achieve the same performance, and thereby measures the sacrifice involved in going to concatenation.

For the moment, we consider only the unexpurgated part of the coding-theorem bound, both for the raw channel and for the superchannel, and we assume that the inner decoder forwards no reliability information with its choice. Then there exists a code of length N and rate R for the raw channel such that the superchannel will have e^{NR} inputs, e^{NR} outputs, and a transition probability matrix p_{ji} for which

$$\Pr(e) = e^{-NR} \sum_{i} \sum_{j \le i} p_{ji} \le e^{-NE(R)}. \tag{99}$$

Applying the unexpurgated part of the coding theorem bound[5] to this superchannel, we can assert the existence of a code of length n and dimensionless rate r (thus rate $r \ln (e^{NR}) = rNR$) which satisfies

$$\Pr(e) \le e^{-nE(r, p_{ji})},$$

where

$$E(r, p_{ji}) \equiv \max_{\substack{\vec{P} \\ 0<\rho\le 1}} \{E_\rho(\vec{P}, p_{ji}) - \rho r^{NR}\}$$

and

$$E_\rho(\vec{P}, p_{ji}) \equiv -\ln \sum_{j} \left[\sum_{i} \vec{P}_i p_{ji}^{\frac{1}{1+\rho}} \right]^{1+\rho}.$$

We cannot proceed with the computation, since we know no more about the matrix p_{ji} than is implied by Eq. 99. We shall now show, however, that of all transition probability matrices satisfying (99), none has smaller $E(r, p_{ji})$ than the matrix $\tilde{p}_{ji}$ defined by

$$\tilde{p}_{ji} = \begin{cases} 1 - e^{-NE(R)}, & i = j \\ \dfrac{e^{-NE(R)}}{e^{NR} - 1}, & i \ne j \end{cases}$$

which is the transition probability matrix of the ideal superchannel with e^{NR} inputs, and $Pr(e) = e^{-NE(R)}$. In this sense, the ideal superchannel is quite the opposite of ideal. (In a sense, for a fixed over-all probability of symbol error, the ideal superchannel is the minimax strategy of nature, while the assumption of an ideal superchannel is the corresponding minimax strategy for the engineer.)

First, we need the following lemma, which proves the convexity of $E_\rho(\vec{P},\ \ )$ over the convex space of all transition probability matrices.

LEMMA: If p_{ji} and q_{ji} are two probability matrices of the same dimensionality, for $0 \leq \lambda \leq 1$,

$$\lambda E_\rho(\vec{P}, p_{ji}) + (1-\lambda)\, E_\rho(\vec{P}, q_{ji}) \geq E_\rho(\vec{P}, \lambda p_{ji}+(1-\lambda)q_{ji}).$$

PROOF: The left-hand side of the inequality is

$$\lambda E_\rho(P, p_{ji}) + (1-\lambda)\, E_\rho(P, q_{ji}) = -\lambda \ln \sum_j \left[\sum_i P_i p_{ji}^{\frac{1}{1+\rho}}\right]^{1+\rho} - (1-\lambda) \ln \sum_j \left[\sum_i P_i q_{ji}^{\frac{1}{1+\rho}}\right]^{1+\rho}$$

$$= -\ln \left[\sum_j \left(\sum_i P_i p_{ji}^{\frac{1}{1+\rho}}\right)^{1+\rho}\right]^{\lambda} \left[\sum_j \left(\sum_i P_i q_{ji}^{\frac{1}{1+\rho}}\right)^{1+\rho}\right]^{1-\lambda}$$

$$\equiv -\ln L,$$

while the right is

$$E_\rho(\vec{P}, \lambda p_{ji}+(1-\lambda)q_{ji}) = -\ln \sum_j \left[\sum_i P_i(\lambda p_{ji}+(1-\lambda)q_{ji})^{\frac{1}{1+\rho}}\right]^{1+\rho} \equiv -\ln R.$$

But

$$L \leq \lambda \sum_j \left[\sum_i P_i p_{ji}^{\frac{1}{1+\rho}}\right]^{1+\rho} + (1-\lambda) \sum_j \left[\sum_i P_i q_{ji}^{\frac{1}{1+\rho}}\right]^{1+\rho}$$

$$= \sum_j \left[\sum_i P_i(\lambda p_{ji})^{\frac{1}{1+\rho}}\right]^{1+\rho} + \left[\sum_i P_i((1-\lambda)q_{ji})^{\frac{1}{1+\rho}}\right]^{1+\rho}$$

$$\leq \sum_j \left[\sum_i P_i(\lambda p_{ji}+(1-\lambda)q_{ji})^{\frac{1}{1+\rho}}\right]^{1+\rho} = R,$$

where the first inequality is that between the arithmetic and geometric means, and the

second is Minkowski's inequality, which is valid because $0 < 1/1+\rho \leq 1$. But if $L \leq R$, $-\ln L \geq -\ln R$, so the lemma is proved.

From this lemma one can deduce by induction that $\overline{E_\rho(\vec{P}, p_{ji})} \geq E_\rho(\vec{P}, \overline{p_{ji}})$, where the bar indicates an average over any ensemble of transition probability matrices. The desired theorem follows.

THEOREM: If $e^{-NR} \sum_i \sum_{j \neq i} p_{ji} \equiv K \leq e^{-NE(R)}$, then

$$E(r, p_{ji}) \geq E(r, \tilde{p}_{ji}),$$

where

$$\tilde{p}_{ii} = 1 - e^{-NE(R)}, \qquad \text{all } i$$

$$\tilde{p}_{ji} = \frac{e^{-NE(R)}}{e^{NR} - 1}, \qquad i \neq j.$$

PROOF: Let $\overrightarrow{e^{-NR}}$ be the particular assignment $\vec{P}$ in which $P_i = e^{-NR}$, all i, which because of its symmetry is clearly the optimum assignment for the ideal superchannel. Then

$$E(r, p_{ji}) = \max_{\rho, \vec{P}} E_\rho(\vec{P}, p_{ji}) - \rho r NR$$

$$\geq E_\rho\left(\overrightarrow{e^{-NR}}, p_{ji}\right) - \rho r NR, \qquad 0 < \rho \leq 1.$$

Suppose we permute the inputs and outputs so that the one-to-one correspondence between them is maintained, thereby getting a new matrix P', for which evidently $E_\rho\left(\overrightarrow{e^{-NR}}, p_{ji}\right) = E_\rho\left(\overrightarrow{e^{-NR}}, p'_{ji}\right)$. Averaging over the ensemble of all $(e^{NR})!$ such permutations, and noting that

$$\overline{p_{ii}} = 1 - K, \qquad \text{all } i$$

$$\overline{p_{ji}} = \frac{K}{e^{NR} - 1}, \qquad i \neq j,$$

we have

$$E_\rho\left(\overrightarrow{e^{-NR}}, \overline{p_{ji}}\right) \leq \overline{E_\rho\left(\overrightarrow{e^{-NR}}, p_{ji}\right)} = E_\rho\left(\overrightarrow{e^{-NR}}, p_{ji}\right).$$

Obviously, $E_\rho\left(\overrightarrow{e^{-NR}}, \tilde{p}_{ji}\right) \leq E_\rho\left(\overrightarrow{e^{-NR}}, \overline{p_{ji}}\right)$, since $K \leq e^{-NE(R)}$, so that finally

$$E(r, p_{ji}) \geq \max_{0<\rho\leq 1} E_\rho\left(\overrightarrow{e^{-NR}}, \tilde{p}_{ji}\right) - \rho r NR = E(r, \tilde{p}_{ji}).$$

In Section II we computed the error exponent for this case, and found that

$$\Pr(e) \leq e^{-nNE^*(r,R)},$$

where

$$E^*(r, R) = (1-r) \min \{R, E(R)\}. \qquad (100)$$

To get the tightest bound for a fixed over-all rate R', we maximize $E^*(r, R)$ subject to the constraint $rR = R'$. Let us define R_E to be the R satisfying $R_E = E(R_E)$; clearly, we never want $R < R_E$, so that $E_C(R')$ can be expressed

$$E_C(R') = \max_{\substack{rR=R' \\ R \geq R_E}} E(R)(1-r). \qquad (101)$$

The computational results of Section VI suggest that the r and R maximizing this expression are good approximations to the rates that are best used in the concatenation of BCH codes.

Geometrically, we can visualize how $E_C(R')$ is related to E(R) in the following way (see Fig. 9). Consider Eq. 100 in terms of R' for a fixed R:

$$E_R^*(R') = \left(1 - \frac{R'}{R}\right) \min \{R, E(R)\}.$$

This is a linear function of R' which equals zero at $R' = R$ and equals $\min\{R, E(R)\}$ at $R' = 0$. In Fig. 9 we have sketched this function for $R = R_1$, R_2, and R_3 greater than R_E, for R_E, and for R_4 less than R_E. $E_C(R')$ may be visualized as the upper envelope of all these functions.

As R' goes to zero, the maximization of (101) is achieved by $R = R_E$, $r \to 0$, so that

$$E_C(0) = E(R_E) = R_E.$$

Since the E(R) curve lies between the two straight lines $L_1 = E(0)$ and $L_2 = E(0) - R$, we have

$$E(0) \geq E(R_E) \geq E(0) - R_E$$

or

$$E(0) \geq E(R_C) \geq \frac{1}{2} E(0).$$

The efficiency $\eta(0) = E_C(0)/E(0)$ is therefore between one-half and one at $R' = 0$.

As R' goes to the capacity C, $E_C(R')$ remains greater than zero for all $R' < C$, but the efficiency approaches zero. For, let $E(R) = K(C-R)^2$ near capacity, which is the normal case (and is not essential to the argument). Let $R' = C(1-\epsilon)$, $\epsilon > 0$; the maximum of (101) occurs at $R = C(1-2\epsilon/3)$, where $E_C(R) = 4\epsilon^3KC^2/27 > 0$. Hence $\eta(R') = 4\epsilon/27$, so that the efficiency goes to zero as R' goes to C. The efficiency is proportional to $(1-R'/C)$, however, which indicates that the drop-off is not precipitous. Most

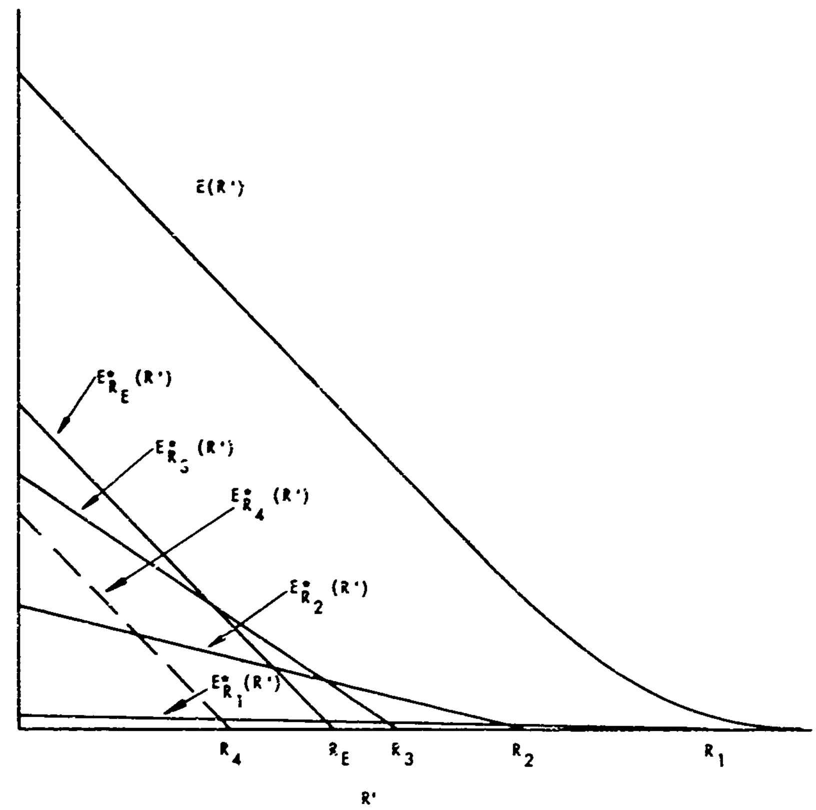

Fig. 9. Derivation of $E_C(R')$ from ECR'.

important, this makes the coding theorem so provocative that exponential decrease in Pr(e) at all rates below capacity is preserved.

We know from the previous discussion that over that part of the curved segment of $E_C(R')$ for which $r > 1/2$, which will normally be [when $E(R_E)$ is on the straight-line segment of $E(R)$] the entire curved segment, Reed-Solomon codes are capable of achieving the error exponent $E_C(R')$ if we use maximum-likelihood decoding. If we use minimum-distance decoding, then we can achieve only

$$\Pr(e) \leq e^{-nNE_m(R')},$$

where

$$E_m(R') = \max_{rR=R'} E(R)(1-r)/2.$$

Over the curved segment of $E_C(R)$, therefore, $E_m(R')$ is one-half of $E_C(R')$; below this segment $E_m(R')$ will be greater than $E_C(R')/2$, and, in fact, for $R' = 0$

$$E_m(0) = E(0)/2$$

which will normally equal $E_C(0)$. Thus minimum-distance decoding costs us a further factor of one-half or better in efficiency, but, given the large sacrifice in efficiency already made in going to concatenated codes, this further sacrifice seems a small enough price to pay for the great simplicity of minimum-distance decoding.

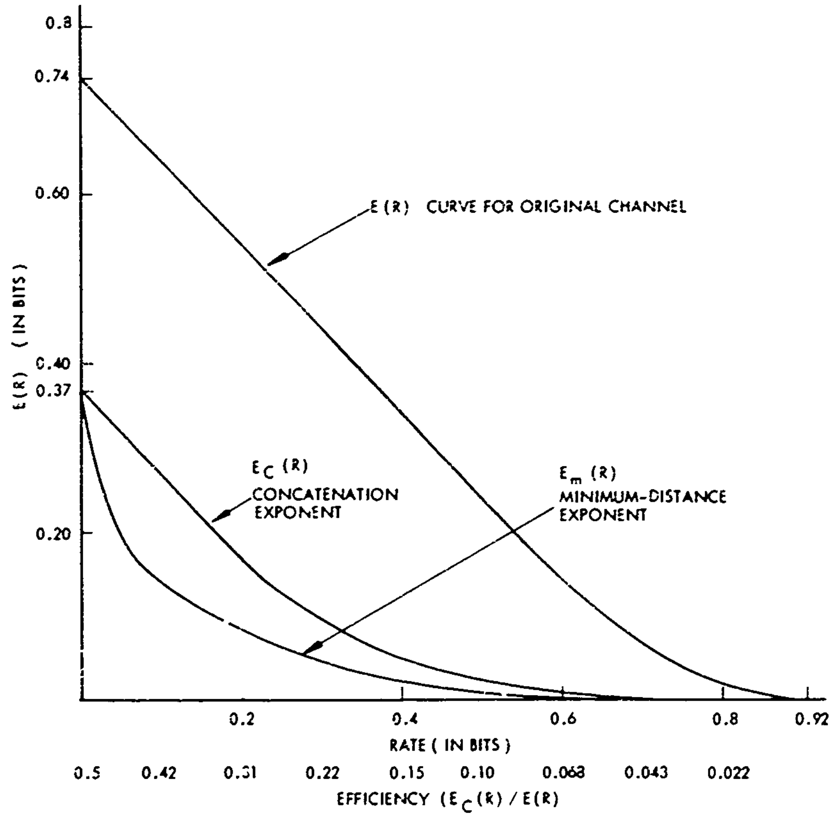

Fig. 10. E(R) curve for original channel.

In Fig. 10 we plot the concatenated exponent $E_C(R')$, the minimum-distance exponent $E_m(R')$, and the original error exponent $E(R')$ of a binary symmetric channel with crossover probability .01. The efficiency ranges from 1/2 to approximately .02 at 9/10 of capacity, which indicates that concatenated codes must be from 2 to 50 times longer than unconcatenated. We shall find that these efficiencies are roughly those obtained in the concatenation of BCH codes.

It is clear that in going to a great number of stages, the error exponent approaches zero everywhere, as we would expect.

We have not considered the expurgated part of the coding-theorem bound for two reasons: first, we are usually not interested in concatenating unless we want to signal at high rates, for which complex schemes are required; second, a lemma for the expurgated bound similar to our earlier lemma is lacking, so that we are not sure the ideal superchannel is the worst of all possible channels for this range. Assuming such a lemma, we then find nothing essentially new in this range; in particular, $\eta(0)$ remains equal to 1/2.

Finally, let us suppose that the inner decoder has the option of making deletions. Since all deletions are equivalent, we lump them into a single output, so that now the superchannel has e^{NR} inputs and $1 + e^{NR}$ outputs. Let the error probability for the superchannel be e^{-NE} and the deletion probability e^{-ND}; assuming the ideal

superchannel with deletions again the worst, we have

$$\Pr(e) \leq e^{-nE(r)},$$

where

$$E(r) = \max_{\rho, \vec{P}} E_\rho(\vec{P}) - \rho NRr$$

$$= \max_{0<\rho\leq 1} E_\rho(\overline{e^{-NR}}) - \rho rNR$$

and

$$E_\rho(\overrightarrow{e^{-NR}}) = -\ln\left\{e^{NR}\left[e^{-NR}(1-e^{-NE}-e^{-ND})^{\frac{1}{1+\rho}} + e^{-NR}(e^{NR}-1)^{\frac{\rho}{1+\rho}}\, e^{-\frac{NE}{1+\rho}}\right]^{1+\rho} + e^{-ND}\right\}.$$

As $N \to \infty$, $E_\rho(\overrightarrow{e^{-NR}}) \to \min(E, D, \rho R)$. But, by adding deletion capability, we can only increase the probability of getting either a deletion or an error, so that

$$e^{-NE(R)} \leq e^{-NE} + e^{-ND}$$

and thus $\min(D, E) \geq E(R)$, so that

$$\min(D, E, \rho R) \geq \min(E(R), \rho R).$$

Thus a deletion capability cannot improve the concatenation exponent $E_C(R')$, although it can, of course, bring the minimum-distance exponent $E_m(R')$ closer to $E_C(R')$, and thereby lessen the necessary block length by a factor less than two.

VI. COMPUTATIONAL PROGRAM

The theoretical results that we have obtained are suggestive; however, what we really want to know is how best to design a communication system to meet a specified standard of performance. The difficulty of establishing meaningful measures of complexity forces us to the computational program described here.

6.1 CODING FOR DISCRETE MEMORYLESS CHANNELS

We first investigate the problem of coding for a memoryless channel for which the modulation and demodulation have already been specified, so that what we see is a channel with q inputs, q outputs, and probability of error p. If we are given a desired over-all rate R' and over-all probability of decoding error Pr(e), we set ourselves the task of constructing a list of different coding schemes with rate R' and probability of decoding error upperbounded by Pr(e).

The types of coding schemes which we contemplate are the following. We could use a single BCH code on GF(q) with errors-only minimum-distance decoding. Or, we could concatenate an RS outer code in any convenient field with an inner BCH code. In the latter case, the RS decoder could be set for errors-only or modified deletions-and-errors decoding (cf. sec. 4.6b); we do not consider generalized minimum-distance decoding, because of the difficulty of getting the appropriate probability bounds. If the outer decoder is set for errors-only decoding, the inner decoder is set to correct as many errors as it can, and any uncorrected word is treated by the outer decoder as an error. If the outer decoder can correct deletions, however, the inner decoder is set to correct only up to t_1 errors, where t_1 may be less than the maximum correctable number t_o, and uncorrected words are treated by the outer decoder as deletions.

Formulas for computing the various probabilities involved are derived and discussed in Appendix B. In general, we are successful in finding formulas that are both valid upper bounds and good approximations to the exact probabilities required. The only exception is the formula for computing the probability of undetected error in the inner decoder, when the inner decoder has the option of deletions, where the lack of good bounds on the distribution of weights in BCH codes causes us to settle for a valid upper bound, but not a good approximation.

Within this class of possible schemes, we restrict our attention to a set of 'good' codes. Tables 1-6 are representative of such lists. Tables 1-4 concern a binary symmetric channel with p = .01; the specifications considered are $Pr(e) = 10^{-12}$ for Tables 1-3, $Pr(e) = 10^{-6}$ for Table 4, R' = .5 for Table 1, .7 for Tables 2 and 4, and .8 for Table 3. (For this channel C = .92 bits and $R_{comp} = .74$.) Table 5 concerns a binary symmetric channel with p = .1 (so that C = .53 and $R_{comp} = .32$); the specifications are R' = .15 and $Pr(e) = 10^{-6}$. Table 6 concerns a 32-ary channel with p = .01 (so that C = 4.86 and $R_{comp} = 4.11$); the specifications are R' = 4, and $Pr(e) = 10^{-12}$.

Since the value of a particular scheme depends strongly upon details of implementation

Table 1. Codes of rate .5 that achieve $Pr(e) \leq 10^{-12}$ on a binary symmetric channel with crossover probability p = .01.

(N,K)	D	T	(n,k)	d	t	Nn	Comment
(414,207)	51	25	...			414	one stage
(15,11)	3	1	(76,52)	25	12	1140	e-o
(31,21)	5	2	(69,51)	19	9	2139	e-o
(63,36)	11	5	(48,42)	7	3	3024	'best' e-o
(63,39)	9	4	(52,42)	11	5	3276	e-o
(63,45)	7	3	(54,38)	17	8	3402	e-o
(127,71)	19	9	(38,34)	5	2	4826	e-o
(127,78)	15	7	(33,27)	7	3	4191	e-o
(127,85)	13	6	(32,24)	9	4	4064	e-o
(127,92)	11	5	(46,32)	15	7	5842	e-o
(127,99)	9	4	(62,40)	23	11	7874	e-o
(31,20)	6	2	(45,35)	11	5	1364	d&e
(31,21)	5	1	(77,57)	21	4	2387	d&e
(63,36)	11	4	(40,35)	6	2	2520	d&e
(63,36)	11	3	(72,63)	10	1	4536	d&e
(63,38)	10	4	(41,34)	8	3	2583	d&e
(63,38)	10	3	(47,39)	9	2	4536	d&e
(63,39)	9	3	(42,34)	9	4	2646	d&e

Notes – Tables 1-6

N(n) = length of inner (outer) code
K(k) = number of information digits
D(d) = minimum distance (d – 1 is the number of deletions corrected)
T(t) = maximum number of errors corrected
nN = over-all block length
Comment: e-o = errors-only, d&e = deletions-and-errors decoding in the outer decoder.

Table 2. Codes of rate .7 that achieve $Pr(e) \leq 10^{-12}$ on a binary symmetric channel with crossover probability p = .01.

(N,K)	D	T	(n,k)	d	t	nN	Comment
(2740,1918)	143	71	...			2740	one stage
(127,99)	9	4	(530,476)	55	27	67310	e-o
(255,207)	13	6	(465,401)	65	32	118575	e-o
(255,199)	15	7	(292,262)	31	15	74460	e-o
(255,191)	17	8	(306,286)	21	10	78030	e-o
(255,187)	19	9	(308,294)	15	7	78540	'best' e-o
(127,98)	10	4	(324,294)	31	12	41148	d&e
(127,92)	11	4	(1277,1234)	43	5	162179	d&e
(127,91)	12	5	(1084,1059)	25	10	137668	d&e
(255,199)	15	6	(214,192)	23	4	54570	d&e
(255,198)	16	6	(234,211)	24	3	59670	d&e
(255,192)	16	7	(214,193)	22	9	54570	d&e
(255,191)	17	7	(214,200)	15	3	54570	d&e
(255,190)	18	7	(232,218)	15	3	59160	d&e
(255,190)	18	8	(232,218)	15	7	59160	d&e
(255,187)	19	8	(198,189)	10	3	50490	d&e
(255,186)	20	8	(224,215)	10	2	57120	d&e

Table 3. Codes of rate .8 that achieve $Pr(e) \leq 10^{-12}$ on a binary symmetric channel with crossover probability p = .01.

(N,K)	D	T	(n,k)	d	t	nN	Comment
no single-stage code							
(2047,1695)	67	33	(1949,1883)	67	33	3989603	e-o
(2047,1684)	69	34	(1670,1624)	47	23	3418490	'best' e-o
(2047,1673)	71	35	(1702,1666)	37	18	3483994	e-o
(2047,1662)	73	36	(2044,2014)	31	15	4184068	e-o
(2047,1695)	67	31	(1477,1427)	51	3	3023419	d&e
(2047,1695)	67	32	(866,856)	31	6	1813642	d&e
(2047,1684)	69	32	(1234,1200)	35	3	2525998	d&e
(2047,1684)	69	33	(763,742)	22	5	1561861	d&e
(2047,1673)	71	34	(804,787)	18	5	1645788	d&e

Table 4. Codes of rate .7 that achieve $Pr(e) < 10^{-6}$ on a binary symmetric channel with crossover probability p = .01.

(N,K)	D	T	(n,k)	d	t	nN	Comment
(784,549)	49	24	...			784	one stage
(127,99)	9	4	(236,212)	25	12	29972	e-o
(127,93)	11	5	(475,459)	17	8	60325	e-o
(255,207)	13	6	(204,176)	29	14	52020	e-o
(255,199)	15	7	(136,122)	15	7	34680	e-o
(255,191)	17	8	(123,115)	9	4	31365	"best" e-o
(255,187)	19	9	(132,126)	7	3	33660	e-o
(127,98)	10	4	(564,545)	20	2	71628	d&e
(127,92)	11	4	(140,127)	14	5	17780	d&e
(127,91)	12	5	(477,466)	12	4	60579	d&e
(255,206)	14	6	(128,111)	18	8	32640	d&e
(255,199)	15	6	(98,88)	11	2	24990	d&e
(255,198)	16	6	(102,92)	11	1	26010	d&e
(255,198)	16	7	(92,83)	10	4	23460	d&e
(255,191)	17	7	(92,86)	7	1	23460	d&e
(255,190)	18	7	(100,94)	7	1	25500	d&e
(255,190)	18	8	(100,94)	7	3	25500	d&e
(255,187)	19	8	(88,84)	5	1	22440	d&e
(255,186)	20	8	(100,96)	5	1	25500	d&e

Table 5. Codes of rate .15 that achieve $Pr(e) < 10^{-6}$ on a binary symmetric channel with crossover probability p = .1.

(N,K)	D	T	(n,k)	d	t	nN	Comment
(511,76)	171	85	...			511	one stage
(31,11)	11	5	(59,2	35	17	1829	e-o
(31,6)	15	7	(54,42)	13	6	1674	e-o
(63,18)	21	10	(51,27)	25	12	3213	e-o
(63,16)	23	11	(35,21)	15	7	2205	e-o
(31,11)	11	4	(40,17)	24	5	1240	d&e
(31,10)	12	4	(43,20)	24	4	1333	d&e
(31,10)	12	5	(47,22)	26	10	1457	d&e
(31,6)	15	5	(116,90)	27	2	3596	d&e
(31,6)	5	6	(45,35)	11	3	1395	d&e

Table 6. Codes of rate 4 that achieve $Pr(e) < 10^{-12}$ on a 32-input symmetric channel with probability of error p = .01.

(N,K)	D	T	(n,k)	d	t	nN	Comment
(540,432)	57	28	...			540	one stage
(31,27)	5	2	(393,361)	33	16	12183	e-o (both codes RS)
(31,25)	7	3	(3250,3224)	27	13	100750	e-o
(148,125)	13	6	(341,323)	19	9	50468	e-o
(148,121)	15	7	(652,638)	15	7	96496	e-o
(223,196)	15	7	(245,223)	23	11	54635	e-o
(223,192)	17	8	(198,184)	15	7	44154	e-o
(223,188)	19	9	(196,186)	11	5	43708	e-o
(298,267)	17	8	(243,217)	27	13	72414	e-o
(298,263)	19	9	(172,156)	17	8	51256	e-o
(298,259)	21	10	(151,139)	13	6	44998	e-o
(298,255)	23	11	(123,115)	9	4	36654	e-o
(298,251)	25	12	(120,114)	7	3	35760	e-o
(31,26)	6	2	(434,414)	21	7	13454	d&e
(148,125)	13	5	(266,252)	15	2	39368	d&e
(148,123)	14	6	(375,361)	15	6	55500	d&e
(148,121)	15	6	(466,456)	11	2	68968	d&e
(223,196)	15	6	(168,153)	16	2	37464	d&e
(223,192)	17	7	(128,119)	10	2	28544	d&e
(298,263)	19	8	(107,97)	11	2	31886	d&e
(298,259)	21	9	(89,82)	8	2	26522	d&e

Table 7. $P = 10^{-12}$.

R'	Single-Stage (N,K)	T	Two-Stage (N,K)	T	P_ϵ	(n,k)	t	r	$r_{5.4}$	nN	η	$\eta_{5.4}$
.1	(53,6)	11	(15,5)	3	.00001	(6,2)	2	.33	(.37)	90	.59	(.42)
.3	(178,54)	17	(15,7)	2	.0004	(23,15)	4	.65	(.63)	345	.52	(.22)
.4	(207,83)	18	(31,16)	3	.0002	(36,28)	4	.78	(.74)	1116	.19	(.15)
.5	(414,207)	25	(63,36)	5	.00004	(48,42)	3	.88	(.80)	3024	.14	(.10)
.6	(788,473)	34	(127,85)	6	.0003	(97,87)	5	.90	(.86)	12319	.064	(.068)
.7	(2740,1918)	71	(255,187)	9	.0003	(308,294)	7	.95	(.91)	78540	.035	(.043)
.75	(6552,4914)	130	(511,394)	13	.0007	(880,856)	12	.97	(.93)	449680	.015	(.032)
.8	no code succeeds		(2047,1684)	34	.002	(1670,1624)	23	.97	(.95)	3418490	...	

Table 8. $P = 10^{-6}$.

R'	Single-Stage (N,K)	T	Two-Stage (N,K)	T	P_ϵ	(n,k)	t	r	$r_{5.4}$	nN	η	$\eta_{5.4}$
.3	(30,10)	5	(7,4)	1	.002	(9,5)	2	.56	(.63)	63	.49	(.22)
.4	(94,38)	8	(15,7)	2	.0004	(28,24)	2	.86	(.74)	420	.22	(.15)
.5	(112,56)	9	(31,21)	2	.004	(31,23)	4	.74	(.80)	961	.12	(.10)
.6	(230,138)	12	(63,45)	3	.004	(63,53)	5	.84	(.86)	3969	.058	(.068)
.7	(784,549)	24	(255,191)	8	.001	(123,115)	4	.93	(.91)	31365	.025	(.043)
.75	(1672,1254)	39	(511,403)	12	.002	(286,272)	7	.95	(.93)	146146	.011	(.032)
.8	(8060,6448)	126	(2047,1695)	33	.003	(827,799)	14	.97	(.95)	1692869	.0048	(.022)

Notes – Tables 7 and 8.

R' = over-all rate
P_ϵ = probability of decoding error in inner decoder
r = dimensionless rate of outer code
$r_{5.4}$ = optimum r as calculated in section 5.4
η = length of best single-stage code divided by nN
$\eta_{5.4}$ = predicted efficiency of concatenation from section 5.4
The tables are of 'best' codes, single- and double-stage, that achieve $Pr(e) \leq P$.

and the requirements of a particular system, we cannot say that a particular entry on any of these lists is 'best.' If minimum over-all block length is the overriding criterion, then a single stage of coding is the best solution; however, we see that using only a single stage to achieve certain specifications may require the correction of a great number of errors, so that almost certainly at some point the number of decoding computations becomes prohibitive. Then the savings in number of computations which concatenation affords may be quite striking.

Among the concatenated codes with errors-only decoding in the outer decoder, the 'best' code is not too difficult to identify approximately, since the codes that correct the fewest errors over all tend also to be those with comparatively short block lengths. Tables 7 and 8 display such 'best' codes for a range of rates and $Pr(e) = 10^{-12}$ and 10^{-6}, on a BSC with $p = .01$; the best single-stage codes are also shown for comparison.

a. Discussion

From these tables we may draw a number of conclusions, which we shall now discuss.

From Tables 1-6 we can evaluate the effects of using deletions-and-errors rather than errors-only decoding in the outer decoder. These are

1. negligible effect on the inner code;

2. reduction of the length of the outer code and hence the over-all block length by a factor less than two; and

3. appreciable savings in the number of computations required in the outer decoder.

From comparison of Tables 2 and 4 and of 7 and 8 we find that the effects of squaring the required probability of error, at moderately high rates, are

1. negligible effect on the inner code; and

2. increase of the length of the outer code and hence the over-all block length by a factor greater than two.

We conclude that, at the moderately high rates where concatenation is most useful, the complexity of the inner code is affected only by the rate required, for a given channel.

These conclusions may be understood in the light of the following considerations. Observe the columns in Tables 7 and 8 which tabulate the probability of decoding error for the inner decoder, which is the probability of error in the superchannel seen by the outer decoder. This probability remains within a narrow range, approximately 10^{-3}-10^{-4}, largely independent of the rate or over-all probability of error required. It seems that the only function of the inner code is to bring the probability of error to this level, at a rate slightly above the over-all rate required.

Thus the only relevant question for the design of the inner coder is: How long a block length is required to bring the probability of decoding error down to 10^{-3} or so, at a rate somewhat in excess of the desired rate? If the outer decoder can handle deletions, then we substitute the probability of decoding failure for that of decoding error in this

question, but without greatly affecting the answer, since getting sufficient minimum distance at the desired rate is the crux of the problem.

Once the inner code has achieved this moderate probability of error, the function of the outer code is to drive the over-all probability of error down to the desired value, at a dimensionless rate near one.

The arguments of section 5.4 are a useful guide to understanding these results. Recall that when the probability of error in the superchannel was small, the over-all probability of error was bounded by an expression of the form

$$\Pr(e) \leq e^{-nNE_1(R')}.$$

Once we have made the superchannel probability of error 'small' (apparently $\sim 10^{-3}$), we then achieve the desired over-all probability of error by increasing n. To square the Pr(e), we would expect to have to double n. Actually, n increases by more than a factor of two, which is due to our keeping the inner and outer decoders of comparable complexity.

That the length of the outer code decreases by somewhat less than a factor of two when deletions-and-errors decoding is permitted is entirely in accord with the results of section 5.4. Basically, the reason is that to correct a certain number of deletions requires one-half the number of check digits in the outer code as to correct the same number of errors, so that for a fixed rate and equal probabilities of deletion or error, the deletion corrector will be approximately half as long.

Finally, we observe that, surprisingly, the ratios of the over-all length of a concatenated code of a given rate to that of a single-stage code of the same rate are given qualitatively by the efficiencies computed in section 5.4 – surprisingly, since the bounds of that section were derived by random-coding arguments whereas here we consider BCH codes, and since those bounds are probably not tight. The dimensionless rate of the outer code also agrees approximately with that specified in section 5.4 as optimum for a given over-all rate.

In summary, the considerations of section 5.4 seem to be adequate for qualitative understanding of the performance of concatenated codes on discrete memoryless channels.

6.2 CODING FOR A GAUSSIAN CHANNEL

We shall now take up the problem of coding for a white additive Gaussian noise channel with no bandwidth restrictions, as an example of a situation in which we have some freedom in choosing how to modulate the channel.

One feasible and near-optimum modulation scheme is to send one of $M \equiv 2^{R_o}$ biorthogonal waveforms every T seconds over the channel. (Two waveforms are orthogonal if their crosscorrelation is zero; a set of waveforms is biorthogonal if it consists of M/2 orthogonal waveforms and their negatives.) If every waveform has energy S, and

the Gaussian noise has two-sided spectral density $N_o/2$, then we say the power signal-to-noise ratio is S/N_oT. Since the information in any transmission is R_o bits, the information rate is R_o/T bits per second; finally, we have the fact that the dimensionless quantity <u>signal-to-noise ratio per information bit</u> is $S/(N_oR_o)$.

$S/(N_oR_o)$ is commonly taken as the criterion of efficiency for signaling over unlimited bandwidth white Gaussian noise channels. Coding theorem arguments[39] show that for reliable communication it must exceed $\ln 2 \sim .7$. Our objective will be to achieve a given over-all probability of error for fixed $S/(N_oR_o)$, with minimum complexity of instrumentation.

The general optimal method[39] of demodulating and detecting such waveforms is to set up a bank of $M/2$ matched filters. For example, the signals might be orthogonal sinusoids, and the filters narrow-bandpass filters. In some sense, the complexity of the receiver is therefore proportional to the number of matched filters that are required — that is, to M. The bandwidth occupied is also proportional to M.

Another method of generating a set of biorthogonal waveforms, especially interesting for its relevance to the question of the distinction between modulation and coding, is to break the T-second interval into $(2T/M)$-sec subintervals, in each of which either the positive or the negative of a single basic waveform is transmitted. If we make the correspondences (positive $\longleftrightarrow$ 1) and (negative $\longleftrightarrow$ 0), we can let the M sequences be the code words of the $(M/2, R_o)$ binary code that results from adding an over-all parity check to an $(M/2-1, R_o)$ BCH code; it can then be shown that the M waveforms so generated are biorthogonal. If they are detected by matched filters, then we would say that we were dealing with an M-ary modulation scheme. On the other hand, this $(M/2, R_o)$ code can be shown to have minimum distance $M/4$, and is thus suitable for a decoding scheme in which a hard decision on the polarity of each $(2T/M)$-sec pulse is followed by a minimum-distance decoder. In this last case we would say that we were dealing with binary modulation with coding, rather than M-ary modulation as before, though the transmitted signals were identical. The same sequences could be decoded (or detected) by many methods intermediate between these extremes, so finely graded that to distinguish where modulation ends and coding begins could only be an academic exercise.

We use maximum-likelihood decoding for the biorthogonal waveforms; the corresponding decision rule for a matched filter detector is to choose the waveform corresponding to the matched filter whose output at the appropriate sample time is the greatest in magnitude, with the sign of that output. Approximations to the probability of incorrect decision with this rule are discussed in Appendix B. In some cases, we permit the detector not to make a decision — that is, to signal a deletion — when there is no matched filter output having magnitude greater by a threshold D or more than all other outputs; in Appendix B we also discuss the probabilities of deletion and of incorrect decision in this case.

We consider the following possibilities of concatenating coding with M-ary modulation to achieve a specified probability of error and signal-to-noise ratio per information bit.

First, we consider modulation alone, with R_o chosen large enough so the specifications are satisfied. Next, we consider a single stage of coding, with a number of values of R_o, and with both errors-only or deletions-and-errors decoding. (If r is the dimensionless rate of the code, the signal-to-noise ratio per information bit is now $S/(N_oR_or)$.) Finally, we consider two stages of coding, or really three-stage concatenation.

Tables 9-11 are representative of the lists that were obtained. Table 9 gives the results for $S/(N_oR_or) = 5$, $Pr(e) = 10^{-12}$; Table 10 for $S/(N_oR_or) = 2$, $Pr(e) = 10^{-12}$; and Table 11 for $S/(N_oR_or) = 2$, $Pr(e) = 10^{-3}$. Again, one cannot pick unambiguously the 'best' scheme; however, the schemes in which M is large enough so that a single Reed-Solomon code of length less than M can meet the required specifications would seem to be very much the simplest, unless some considerations other than those that we have contemplated heretofore were significant.

To organize our information about these codes, we choose to ask the question: For a fixed M and specified Pr(e), which RS code of length M-1 requires the minimum signal-to-noise ratio per information bit? Tables 12-15 answer this question for $R_o \leq 9$ (after which the computer overflowed), and for $Pr(e) = 10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}$. Except in Table 15, we have considered only errors-only decoding, since Table 15 shows that, even for $Pr(e) = 10^{-12}$, allowing deletions-and-errors decoding improves things very little, to the accuracy of our bounds, and does not affect the character of the results. The $S/(N_oR_o)$ needed to achieve the required probability of error without coding, for $R_o \leq 20$, is also indicated.

a. Discussion

Let us first turn out attention to Table 9, which has the richest selection of diverse schemes, as well as being entirely representative of all of the lists that we generated. Certain similarities to the lists for discrete memoryless channels are immediately evident. For instance, the use of deletions allows some shortening and simplification of the outer decoder, though not as much as before. Also, for fixed M, going to two stages of coding rather than one lessens the computational demands on the decoders, at the price of much increased block length.

It seems clear that it is more efficient to let M become large enough so that two stages of coding are unnecessary, and in fact large enough that a single RS code can be used. As M falls below this size, the needed complexity of the codes would seem to increase much more rapidly than that of the modulation decreases, while for larger M the reverse is true. The explanation is that a certain M is required to drive the probability of detection error down to the point where coding techniques become powerful, for $S/(N_oR_o)$ somewhat less than the final signal-to-noise ratio per information bit. Once this moderate probability has been achieved, it would seem to be wasteful to use modulation techniques to drive it much lower by increasing M. Tables 10 and 11 illustrate this point by showing that this critical M is not greatly affected by an enormous change in required Pr(e).

Table 9. Modulation and coding that achieve $Pr(e) < 10^{-12}$ with a signal-to-noise ratio per information bit of 5, on a Gaussian channel.

M	(N,K)	D	T	(n,k)	d	t	kKR_o	d/b	Comment
16384	...			...			14	571.4	no coding
64	(21,15)	7	3	...			90	7.47	e-o
64	(20,12)	9	4	...			72	8.89	e-o
32	(26,18)	9	4	...			90	4.62	e-o
32	(26,16)	11	5	...			80	5.20	e-o
16	(155,136)	11	5	...			544	2.28	e-o
16	(90,67)	13	6	...			268	2.69	e-o
16	(85,58)	15	7	...			232	2.93	e-o
16	(80,50)	17	8	...			200	3.20	e-o
16	(75,43)	19	9	...			172	3.49	e-o
8	(236,184)	21	10	...			552	1.71	e-o
8	(201,138)	25	12	...			414	1.94	e-o
8	(197,124)	29	14	...			372	2.12	e-o
2	(511,358)	37	18	...			358	1.43	e-o
2	(481,310)	41	20	...			310	1.55	e-o
2	(461,254)	51	25	...			254	1.81	e-o
64	(43,37)	7	1	...			222	6.20	d&e
64	(41,33)	9	1	...			198	6.63	d&e
64	(26,22)	5	2	...			132	6.30	d&e
64	(19,13)	7	2	...			78	7.79	d&e
64	(22,14)	9	2	...			84	8.38	d&e
64	(18,12)	7	3	...			72	8.00	d&e
32	(29,23)	7	2	...			115	4.03	d&e
32	(30,22)	9	2	...			110	4.36	d&e
32	(25,19)	7	3	...			95	4.21	d&e
32	(22,14)	9	3	...			70	5.03	d&e
16	(127,108)	11	3	...			432	2.35	d&e
16	(117,94)	13	3	...			376	2.49	d&e
16	(81,62)	11	4	...			248	2.61	d&e
16	(79,56)	13	4	...			224	2.82	d&e
16	(73,50)	13	6	...			200	2.92	d&e
16	(15,11)	5	2	(25,21)	5	2	924	3.25	e-o
8	(43,36)	5	2	(77,69)	9	4	7452	1.78	e-o
8	(48,37)	7	3	(48,42)	7	3	4662	1.98	e-o
8	(63,49)	9	4	(31,27)	5	2	3969	1.97	e-o
2	(63,45)	7	3	(92,80)	13	6	3600	1.61	e-o
2	(63,39)	9	4	(92,82)	11	5	3198	1.81	e-o
2	(63,36)	11	5	(63,55)	9	4	1980	2.00	e-o

Notes: Tables 9-11.

N, K, D, T, n, k, d, t have been defined in Section I
M = number of biorthogonal signals transmitted
kKR_o = total bits of information in a block
d/b = dimensions required $(nNM/(2kKR_o))$ per information bit.

Table 10. Modulation and coding that achieve $Pr(e) \leq 10^{-12}$ with a signal-to-noise ratio per information bit of 2, on a Gaussian channel.

M	(N,K)	D	T	(n,k)	d	t	Comment
512	(211,167)	45	22	...			e-o
512	(261,209)	43	21	...			e-o
512	(311,271)	41	20	...			e-o
256	(255,195)	61	30	...			e-o
128	(127,97)	31	15	(127,119)	9	4	e-o
128	(127,99)	29	14	(127,117)	11	5	e-o
128	(127,101)	27	13	(127,124)	4	0	d&e
128	(127,104)	24	11	(127,122)	6	0	d&e
128	(127,104)	24	10	(127,120)	8	0	d&e

Note: The special RS bound on weights in section 3.3a has been used to compute probabilities for the last three codes. With the general bound of Appendix B, it appears that deletions are no help.

Table 11. Modulation and coding that achieve $Pr(e) \leq 10^{-3}$ with a signal-to-noise ratio per information bit of 2, on a Gaussian channel.

M	(N,K)	D	T	(n,k)	d	t	Comment
16384	...			...			no coding
256	(37,27)	11	5	...			e-o
256	(45,37)	9	4	...			e-o
128	(48,34)	15	7	...			e-o
128	(50,38)	13	6	...			e-o
64	(895,719)	91	45	...			e-o

Note: Deletions are no help.

Tables 12-15. Minimum $S/(N_oR_or)$ achievable on a Gaussian channel.

Table 12.
$Pr(e) = 10^{-3}$.

R_o	no code	RS code
1	4.78	
2	5.42	
3	4.26	4.23
4	3.57	3.11
5	3.12	2.41
6	2.81	2.02
7	2.59	1.77
8	2.41	1.61
9	2.28	1.50
10	2.16	
11*	2.18	
12*	2.11	
14*	2.00	
16*	1.92	
18*	1.85	
20*	1.80	

Table 13.
$Pr(e) = 10^{-6}$.

t	no code	RS code
	11.30	
	11.96	
1	8.68	7.34
3	6.92	4.59
5	5.83	3.19
9	5.09	2.44
18	4.56	2.01
33	4.16	1.76
62	3.85	1.60
	3.60	
	3.40	
	3.23	
	2.96	
	2.76	
	2.60	
	2.48	

Table 14.
$Pr(e) = 10^{-9}$.

t	no code	RS code	t
	17.98		
	18.66		
1	13.16	10.42	1
3	10.28	6.01	3
5	8.52	3.88	6
10	7.34	2.80	11
19	6.49	2.21	19
34	5.85	1.88	35
64	5.35	1.67	65
	4.95		
	4.63		
	4.35		
	3.93		
	3.61		
	3.36		
	3.16		

Table 15. $Pr(e) = 10^{-12}$.

R	no code	RS code	t	p_ϵ	RS code (d&e)
1	24.74				
2	25.42				
3	17.67	13.53	1	.0000002	13.60
4	13.67	7.45	3	.0001	6.86
5	11.23	4.54	6	.002	4.25
6	9.60	3.13	11	.009	3.02
7	8.43	2.40	20	.02	2.38
8	7.55	1.98	36	.036	
9	6.86	1.73	67	.05	
10	6.31				
11*	5.86				
12*	5.49				
14*	4.90				
16*	4.46				
18*	4.11				
20*	3.84				

Notes: Tables 12-15.

$R_o = \log_2 M$

no code = minimum signal-to-noise ratio per information bit achievable without coding

RS code = minimum signal-to-noise ratio per information bit achievable with an RS code of length $M-1$

t = number of errors which the RS code must correct

RS code (d&e) = minimum signal-to-noise ratio per information bit achievable by an RS code correcting t errors and 2t deletions.

*For these values of R_o a weaker probability bound was used (see Appendix B).

Since the RS codes are the most efficient of the BCH class with respect to the number of check digits required to achieve a certain minimum distance and hence error-correction capability, another important effect of increasing M is to make the symbol field GF(M) large enough that RS codes of the necessary block lengths can be realized. Once M is large enough to do this, further increases result in no further increase of efficiency in this respect.

Tables 12-15 are presented as much for reference as for a source of further insight. It is interesting to note that for a given M, the same RS code is approximately optimum over a wide range of required Pr(e). No satisfactory explanation for this constancy has been obtained; lest the reader conjecture that there might be some universal optimality to these codes, however, it might be mentioned that the same tables for a different type of probability distribution than the Gaussian show markedly different codes as optimum. Table 15 includes the superchannel probabilities of error seen by the outer coder; they are somewhat higher than the comparable probabilities for the discrete memoryless channel, 10^{-2}-10^{-3}, but remain in the same approximate range.

6.3 SUMMARY

A most interesting conclusion emerges from these calculations. A distinct division of function between the outer code and the inner stages — of modulation, or inner coding, or perhaps both — is quite apparent. The task of the inner stages, while somewhat exceeding the specified rate or $S/(N_oR_o)$, is to turn the raw channel into a superchannel with moderate (10^{-2}-10^{-4}) probability of error, and enough inputs so that an RS code may be used as the outer code. The function of the outer code is then to drive the over-all probability of error as low as desired, at a dimensionless rate close enough to one not to hurt the over-all rate or $S/(N_oR_o)$ badly.

For future work, two separate problems of design are suggested. The first is the most efficient realization of RS encoders and decoders, with which we were concerned in Section IV. The second, which has been less explored, is the problem of efficient realization of a moderate probability of error for given specifications. Communication theory has previously focused largely on the problem of achieving negligibly small probabilities of error, but the existence of RS codes solves this problem whenever the problem of achieving a probability of error less than 10^{-3}, say, can be solved. This last problem is probably better considered from the point of view of modulation theory or signal design than coding theory, whenever the former techniques can be applied to the channel at hand.

APPENDIX A

Variations on the BCH Decoding Algorithm

A.1 ALTERNATIVE DETERMINATION OF ERROR VALUES

The point of view which led us to the erasure correction procedure of section 4.5 leads us also to another method of determining the values of the errors. Suppose the number of errors t has been discovered; then the $t \times t$ matrix M has rank t and therefore nonzero determinant. Let the decoder now determine the locator X_{j_o} of any error. If we were to guess the corresponding error value e_{j_o} and modify the T_ℓ accordingly, the guessed word would still have either t or (on the chance of a correct guess) $t-1$ errors; thus the $t \times t$ matrix M_t' formed from the new T_ℓ' would have zero determinant if and only if the guess were correct. In general one would expect this argument to yield a polynomial in e_{j_o} of degree t as the equation of condition, but because of the special form of M_t this equation is only of first degree, and an explicit formula for e_{j_o} can be obtained.

In symbols, let

$$\vec{S}'_{(m_o+n+s,\, m_o+n)} \equiv \vec{S}_{(m_o+n+s,\, m_o+n)} - e_{j_o} \vec{X}_{j_o(m_o+n+s,\, m_o+n)}.$$

Then

$$T_\ell' \equiv \vec{\sigma}_d \cdot \vec{S}'_{(m_o+n+s,\, m_o+n)} = \vec{\sigma}_d \cdot \vec{S}_{(m_o+n+s,\, m_o+n)} - e_{j_o} \vec{\sigma}_d \cdot \vec{X}_{j_o(m_o+n+s,\, m_o+n)}$$

$$= T_\ell - e_{j_o} X_{j_o}^{m_o+n} \sigma_d\left(X_{j_o}\right) = T_\ell - E_{j_o} X_{j_o}^{n}.$$

$$M_t' = \begin{bmatrix} T_{2t_o-2} - E_{j_o} X_{j_o}^{2t_o-2} & T_{2t_o-3} - E_{j_o} X_{j_o}^{2t_o-3} & \cdots & T_{2t_o-t-1} - E_{j_o} X_{j_o}^{2t_o-t-1} \\ T_{2t_o-3} - E_{j_o} X_{j_o}^{2t_o-3} & T_{2t_o-4} - E_{j_o} X_{j_o}^{2t_o-4} & \cdots & T_{2t_o-t-2} - E_{j_o} X_{j_o}^{2t_o-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_o-t-1} - E_{j_o} X_{j_o}^{2t_o-t-1} & T_{2t_o-t-2} - E_{j_o} X_{j_o}^{2t_o-t-2} & \cdots & T_{2t_o-2t} - E_{j_o} X_{j_o}^{2t_o-2t} \end{bmatrix}$$

Let us expand this determinant into 2^t determinants, using the fact that the determinant of the matrix which has the vector $(\vec{a}+\vec{b})$ as a row is the sum of the determinants of the two matrices which have $\vec{a}$ and $\vec{b}$ in that row, respectively. We classify the resulting determinants by the number of rows which have E_{j_o} as a factor.

There is one determinant with no row containing E_{j_o}, which is simply $|M_t|$.

There are t determinants with one row having E_{j_o} as a factor. For example, the first is

$$\begin{bmatrix} -E_{j_o} X_{j_o}^{2t_o-2} & -E_{j_o} X_{j_o}^{2t_o-3} & \cdots & -E_{j_o} X_{j_o}^{2t_o-t-1} \\ T_{2t_o-3} & T_{2t_o-4} & \cdots & T_{2t_o-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_o-t-1} & T_{2t_o-t-2} & \cdots & T_{2t_o-2t} \end{bmatrix}$$

There are $\binom{t}{2}$ determinants with two rows having E_{j_o} as a factor. The first is

$$\begin{bmatrix} -E_{j_o} X_{j_o}^{2t_o-2} & -E_{j_o} X_{j_o}^{2t_o-3} & \cdots & -E_{j_o} X_{j_o}^{2t_o-t-1} \\ -E_{j_o} X_{j_o}^{2t_o-3} & -E_{j_o} X_{j_o}^{2t_o-4} & \cdots & -E_{j_o} X_{j_o}^{2t_o-t-2} \\ T_{2t_o-4} & T_{2t_o-5} & \cdots & T_{2t_o-t-3} \\ \vdots & \vdots & & \vdots \\ T_{2t_o-t-1} & T_{2t_o-t-2} & \cdots & T_{2t_o-2t} \end{bmatrix}$$

But in this determinant the first row is simply X_{j_o} times the second, so that the determinant is zero. Furthermore, in all such determinants with two or more rows having E_{j_o} as a factor, these rows will be some power of X_{j_o} times each other, so that all such determinants are zero.

The t determinants with one row having E_{j_o} as a factor are all linear in E_{j_o}, and contain explicit powers of X_{j_o} between $2t_o - 2t$ and $2t_o - 2$; their sum is then

$$-E_{j_o} X_{j_o}^{2t_o-2t} P\left(X_{j_o}\right)$$

where $P\left(X_{j_o}\right)$ is a polynomial of degree $2t - 2$, whose coefficients are functions of the original T_n.

Finally, we recall that $E_{j_o} = e_{j_o} X_{j_o}^{m_o} \sigma_d\left(X_{j_o}\right)$ and that $|M_t'| = 0$ if and only if e_{j_o} is chosen correctly, from which we get the equation of condition

$$0 = |M_t'| = |M_t| - E_{j_o} X_{j_o}^{2t_o-2t} P\left(X_{j_o}\right)$$

so

$$e_{j_o} = \frac{|M_t|}{X_{j_o}^{m_o+2t_o-2t}\,\sigma_d(X_{j_o})\,P(X_{j_o})}. \tag{A.1}$$

$|M_t|$ can easily be obtained as a by-product of the reduction of M. The only term in the denominator of (A. 1) that is not readily calculable is $P(X_{j_o})$. In general, if A_{ik} is the determinant of the matrix remaining after the i^{th} row and k^{th} column are struck from M_t, then

$$P(X_{j_o}) = \sum_{\ell=2}^{2t} \left(-X_{j_o}\right)^{2t-\ell} \sum_{i+k=\ell} A_{ik}.$$

A simplification occurs when we are in a field of characteristic two. For note that because of the diagonal symmetry of M_t, $A_{ik} = A_{ki}$. Any sum $\sum_{i+k=\ell} A_{ik}$ will consist entirely of pairs $A_{ik} + A_{ki} = 0$, unless ℓ is even, when the entire sum equals A_{jj}, where $j = \ell/2$. Then

$$P(X_{j_o}) = \sum_{j=1}^{t} X_{j_o}^{2(t-j)} A_{jj}.$$

Evaluation of the coefficients of P(X) in a field of characteristic two therefore involves calculating t (t−1) × (t−1) determinants.

A. 11 Example

Let the decoder have solved Eqs. 50 as before, obtaining as a by-product $|M_t| = \alpha^6$. Trivially,

$$A_{22} = T_4 = \alpha^{13}, \quad A_{11} = T_2 = 0.$$

The first error locator that it will discover is $X_1 = \alpha^{14}$. Then, from Eq. A. 1,

$$e_1 = \frac{|M_2|}{X_1^3\left(X_1^2+\sigma_{d1}X_1+\sigma_{d2}\right)\left(A_{11}X_1^2+A_{22}\right)} = \frac{\alpha^6}{\alpha^{12}(\alpha^{13}+\alpha\cdot\alpha^{14}+\alpha^{10})\,\alpha^{13}} = \alpha^4.$$

Similarly, when it discovers $X_2 = \alpha^{11}$,

$$e_2 = \frac{\alpha^6}{\alpha^3(\alpha^7+\alpha\cdot\alpha^{11}+\alpha^{10})\,\alpha^{13}} = \alpha.$$

Then it can solve for d_1 and d_2 as before.

A. 12 Remarks

The procedure just described for determining error values is clearly applicable in principle to the determination of erasure values. In the last case, however, $\vec{\sigma_d}$ must be

replaced by $\overrightarrow{_{k_0}\sigma_d}$, the vector of elementary symmetric functions of the $s-1$ erasures other than the one being considered, and the original modified cyclic parity checks T_ℓ by the modified cyclic parity checks defined on the other $s-1$ erasure locators. This means that the determinants appearing in Eq. A.2, as well as $|M_t|$, must be recomputed to solve for each erasure. In contrast to the solution for the error values, this promises to be tedious and to militate against this method in practice. We mention this possibility only because it does allow calculation of the correct value of an erasure, given only the number of errors and the positions of the other erasures, without knowledge of the location or value of the errors, a capability which might be useful in some application.

The erasure-correction scheme with no errors (section 4.5) can be seen to be a special case of this algorithm.

A.13 Implementation

After we have located the errors, we have the option of solving for the error values directly by (A.1), or indirectly, by treating the errors as erasures and using Eq. 50.

If we choose the former method, we need the t $(t-1)\times(t-1)$ determinants A_{jj} of (A.2). In general this requires

$$\frac{1}{4}t\binom{2t}{3} < \frac{t^4}{3}$$

multiplications, which is rapidly too many as t becomes large. There is a method of calculating all A_{jj} at once which seems feasible for moderate values of t. We assume a field of characteristic two.

Let $B_{a_1,a_2,\ldots,a_j}$ be the determinant of the $j\times j$ matrix which remains when all the rows and columns but the a_1^{th}, a_2^{th}, ..., a_j^{th} are struck from M_t. In this notation

$$|M_t| = B_{1,2,\ldots,t} \quad \text{and} \quad A_{jj} = B_{1,2,\ldots,j-1,j+1,\ldots,t}.$$

The reader, by expanding B in terms of the minors of its last row and cancelling those terms which because of symmetry appear twice, may verify the fact that

$$B_{a_1,a_2,\ldots a_j} = T_{2t_o-2a_j}B_{a_1,a_2,\ldots a_{j-1}} + T^2_{2t_o-2a_j+1}B_{a_1,a_2,\ldots a_{j-2}}$$
$$+ T^2_{2t_o-2a_j+2}B_{a_1,a_2,\ldots a_{j-3},a_{j-1}} + \ldots$$

The use of this recursion relation allows calculation of all A_{jj} with N_t multiplications (not counting squares), where, for small t, N_t is $N_2 = 0$ (see section A.11), $N_3 = 3$, $N_4 = 15$, $N_5 = 38$, $N_6 = 86$, $N_7 = 172$, $N_8 = 333$, $N_9 = 616$.

Once the A_{jj} are obtained, the denominator of (A.1) can be expressed as a single polynomial $E(X)$ by st multiplications; $E(X)$ has terms in X^m, $m_o + 2t_o - 2t \leq m \leq m_o + 2t + s$, or a total of $2t+s+1$ terms. The value of $E(X)$ can therefore be obtained for

$X = 1, \beta^{-1}, \beta^{-2}, \ldots$ in turn by the Chien[31] method of solving for the roots of $\sigma_e(X)$, and in fact these two calculations may be done simultaneously. Whenever β^{n-i} is a root of $\sigma_e(X)$, $E(\beta^{n-i})$ will appear as the current value of $E(X)$. Since $|M_t|$ will have been obtained as a by-product of solving for $\sigma_e(X)$, an inversion and a multiplication will give the error value corresponding to $X_{j_o} = \beta^{n-i}$. Other $n(s+2t)$ multiplications by β^m are involved here, and $s+2t$ memory registers.

In order to compare the alternative methods of finding error values, we simply compare the number of multiplications needed in each case, leaving aside all analysis of any other equipment or operations needed to realize either algorithm. We recall that the values of s erasures can be determined with approximately $2s(s-1)$ multiplications. For the first method, we need approximately N_t multiplications to find the error values, and and $2s(s-1)$ to find the erasures; for the second, $2(s+t)(s+t-1)$ to find both the erasures and the errors. Using the values of N_t given earlier, we find that the former method requires fewer multiplications when $t \leq 7$, which suggests that it ought to be considered whenever the minimum distance of the code is 15 or less.

A.2 ALTERNATIVE DETERMINATION OF ERROR LOCATIONS

Continued development of the point of view expressed above gives us an alternative method of locating the errors. If we tentatively consider a received symbol as an erasure, in a received word with t errors, then the resulting word has t errors if the trial symbol was in error. The vanishing of the $t \times t$ determinant M'' formed from the T''_ℓ defined now by $s+1$ erasure locators then indicates the error locations. The reader may verify the fact that if X_{j_o} is the locator of the trial symbol,

$$T''_\ell = T_{\ell+1} - X_{j_o} T_\ell,$$

and

$$M''_t = \begin{bmatrix} T_{2t_o-1} - X_{j_o} T_{2t_o-2} & T_{2t_o-2} - X_{j_o} T_{2t_o-3} & \cdots & T_{2t_o-t} - X_{j_o} T_{2t_o-t-1} \\ T_{2t_o-2} - X_{j_o} T_{2t_o-3} & T_{2t_o-3} - X_{j_o} T_{2t_o-4} & \cdots & T_{2t_o-t-1} - X_{j_o} T_{2t_o-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_o-t} - X_{j_o} T_{2t_o-t-1} & T_{2t_o-t-1} - X_{j_o} T_{2t_o-t-2} & \cdots & T_{2t_o-2t+1} - X_{j_o} T_{2t_o-2t} \end{bmatrix}$$

If we expand $|M''_t|$ by columns, many of the resulting determinants will have one column equal to $-X_{j_o}$ times another. The only ones that will not will be

$$D_o \equiv \left| \vec{T}_{(2t_o-1, 2t_o-t)}, \vec{T}_{(2t_o-2, 2t_o-t-1)}, \ldots, \vec{T}_{(2t_o-t, 2t_o-2t+1)} \right|$$

$$-X_{j_o} D_1 \equiv \left| \vec{T}_{(2t_o-1, 2t_o-t)}, \ldots, \vec{T}_{(2t_o-t+1, 2t_o-2t+2)}, -X_{j_o} \vec{T}_{(2t_o-t-1, 2t_o-2t)} \right|$$

$$X_{j_o}^2 D_2 \equiv \left| \vec{T}_{(2t_o-1,2t_o-t)}, \ldots, \vec{T}_{(2t_o-t+2,2t_o-2t+3)}, -X_{j_o}\vec{T}_{(2t_o-t,2t_o-2t+1)}, -X_{j_o}\vec{T}_{(2t_o-t-1,2t_o-2t)} \right|$$

and so forth. Thus if X_{j_o} is a root of the polynomial

$$D(X_{j_o}) = \sum_{j=0}^{t} D_j(-X_{j_o})^j,$$

$|M_t^=|$ is zero and X_{j_o} is an error locator. It can be checked by the expansion of D_j into three matrices, as was done earlier in the proof that the rank of M is t, that

$$D_j = \sigma_{e(t-j)} D_t$$

so that

$$D(X) = D_t \sigma_e(X),$$

and this method is entirely equivalent to the former one. Furthermore, it is clear that

$$D(X) = \begin{vmatrix} X^t & T_{2t_o-1} & T_{2t_o-2} & \cdots & T_{2t_o-t} \\ X^{t-1} & T_{2t_o-2} & T_{2t_o-3} & \cdots & T_{2t_o-t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ X & T_{2t_o-t} & T_{2t_o-t-1} & \cdots & T_{2t_o-2t+1} \\ 1 & T_{2t_o-t-1} & T_{2t_o-t-2} & \cdots & T_{2t_o-2t} \end{vmatrix}$$

The condition of the vanishing of this matrix determinant is the generalization to the non-binary case of the 'direct method' of Chien.[31] It appears to offer no advantages in practice, for to get the coefficients of $D(X)$ one must find the determinants of $t+1$ $t \times t$ matrices, whereas the coefficients of the equivalent $\sigma_e(X)$ can be obtained as a by-product of the determination of t.

APPENDIX B

Formulas for Computation

We shall now derive and discuss the formulas used for the computations of Section V.

B.1 OUTER DECODER

Let us consider first the probability of the outer decoder decoding incorrectly, or failing to decode. We shall let p_e be the probability that any symbol is in error, and p_d be the probability that it is erased.

If the outer decoder does errors-only decoding, $p_d = 0$. Let the maximum correctable number of errors be t_o; then the probability of decoding error is the probability of $t_o + 1$ or more symbol errors:

$$\Pr(e) = \sum_{t=t_o+1}^{n} \binom{n}{t} p_e^t (1-p_e)^{n-t}. \tag{B.1}$$

If the outer decoder does deletions-and-errors decoding, the minimum distance is d, and the maximum number of errors corrected is t_o, then the probability of decoding error is the probability that the number of errors t and the number of deletions s satisfy $2t+s \geq d$ or $t \geq t_o + 1$:

$$\Pr(e) = \sum_{t,s} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} \qquad 2t + s \geq d \text{ or } t \geq t_o + 1$$

$$= \sum_{t=0}^{t_o} \sum_{s=d-2t}^{n} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} + \sum_{t=t_o+1}^{n} \binom{n}{t} p_e^t (1-p_e)^{n-t} \tag{B.2}$$

Equation B.2 is also valid for modified deletions-and-errors decoding, when t_o is the reduced maximum correctable number of errors.

For fixed t, we can lower-bound an expression of the form

$$\sum_{s=t_1}^{n} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} \tag{B.3}$$

by

$$\sum_{s=t_1}^{t_2+1} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t}. \tag{B.4}$$

To upperbound (B. 3), we write it as

$$\sum_{s=t_1}^{t_2} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} + \sum_{s=t_2+1}^{n} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t}. \tag{B. 5}$$

Since the ratio of the $(s+1)^{st}$ to the s^{th} term in the latter series is

$$\frac{(n-s-t)p_d}{(s+1)(1-p_e-p_d)} < \frac{(n-t-t_2)p_d}{t_2(1-p_e-p_d)} \equiv a,$$

Eq. B. 5 can be upperbounded by

$$\sum_{s=t_1}^{t_2} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} + \binom{n}{t_2+1,t} p_e^t p_d^{t_2+1} (1-p_e-p_d)^{n-t-t_2-1} \sum_{s'\geq 0} a^{s'} =$$

$$\sum_{s=t_1}^{t_2} \binom{n}{s,t} p_e^t p_d^s (1-p_e-p_d)^{n-s-t} + \frac{1}{1-a} \binom{n}{t_1+1,t} p_e^t p_d^{t_2+1} (1-y_e-p_d)^{n-t-t_2-1} \tag{B. 6}$$

By choosing t_2 large enough, the lower and upper bounds of Eqs. B. 4 and B. 6 may be made as close as desired. In the program of Section V, we let t_2 be large enough so that the bounds were within 1 per cent of each other. Both (B. 1) and (B. 2) can then be upperbounded and approximated by (B. 6).

B. 2 INNER DECODER

If the outer decoder is set to do errors-only decoding, the inner decoder corrects as many errors as it can (t_o). Whenever the actual number of errors exceed t_o, the inner decoder will either fail to decode or decode in error, but either of these events constitutes a symbol error to the outer decoder. If the probability of symbol error for the inner decoder is p_o, then

$$p_e = \sum_{t=t_o+1}^{n} \binom{n}{t} p_o^t (1-p_o)^{n-t}. \tag{B. 7}$$

Equation B. 7 can be upperbounded and approximated by Eq. A. 6.

If the outer decoder is set for deletions-and-errors decoding, the inner decoder is set to correct whenever there are apparently t_1 or fewer errors, where $t_1 \leq t_o$; otherwise it signals a deletion. If there are more than t_1 actual errors, the decoder will either delete or decode incorrectly, so that

$$p_e + p_d = \sum_{t=t_1+1}^{n} \binom{n}{t} p_o^t (1-p_o)^{n-t}.$$

Ordinarily t_1 is set so that $p_e \ll p_d$, so that p_d is upperbounded and approximated by

$$p_d \leq \sum_{t=t_1+1}^{n} \binom{n}{t} p_o^t (1-p_o)^{n-t}, \tag{B.8}$$

which in turn is upperbounded and approximated by Eq. A.6.

Estimating p_e turns out to be a knottier problem. Of course, if the minimum distance of the inner code is d, no error can occur unless the number of symbol errors is at least $d-t$, so that

$$p_e \leq \sum_{t=d-t_1}^{n} \binom{n}{t} p_o^t (1-p)^{n-t}$$

This is a valid upper bound but a very weak estimate of p_e, since in general many fewer than the total of $\binom{n}{t}$ t-error patterns will cause errors; most will cause deletions. A tighter bound for p_e depends, however, on knowledge of the distribution of weights in the inner code, which is in general difficult to calculate.

We can get a weak bound on the number N_w of code words of weight w in any code on GF(q) of length n and minimum distance d as follows. Let t_o be the greatest integer such that $2t_o < d$. The total number of code words of weight $w-t_o$ distance t_o from a code word of weight w is $\binom{w}{t_o}$, since to get such a word we may change any t_o of the w nonzero symbols in the word to zeros. The total number of words of weight $w-t_o$ distance t_o from all code words of weight w is then

$$\binom{w}{t_o} N_w,$$

and all of these are distinct, since no word can be distance t_o from two different code words. But this number cannot exceed the total number of words of weight $w-t_o$:

$$\binom{n}{w-t_o} (q-1)^{w-t_o}.$$

Therefore

$$N_w \leq \frac{n!\, t_o!\, (q-1)^{w-t_o}}{w!\, (n-w-t_o)!}. \tag{B.9}$$

Now a decoding error will occur, when the inner code is linear, when the error

pattern is distance t_1 or less from some code word. The total number of words distance k from some code word of weight w is

$$\sum_{i,j,\ell} \binom{n-w}{\ell}(q-1)^{\ell} \binom{w}{i,j}(q-2)^{i}; \qquad i+j+\ell = k$$

since all code words can be obtained by changing any ℓ of the $n-w$ zeros to any of the $(q-1)$ nonzero elements, any i of the w nonzero elements to any of the other $(q-2)$ nonzero elements, and any j of the remaining nonzero elements to zeros, where $i+j+l=k$. The weight of the resulting word for a particular i, j, l, will be $w+l-j$, so that the probability of getting a word distance k from a particular code word of weight w is

$$\sum_{\substack{i,j,\ell \\ i+j+\ell=k}} \binom{n-w}{\ell}(q-1)^{\ell} \binom{w}{i,j}(q-2)^{i} \left(\frac{p_o}{q-1}\right)^{w+\ell-j} (1-p_o)^{n-w-\ell+j}.$$

Summing over all words of all weights $w \geq d$ and all $k \leq t_1$, and substituting $j = k-i-l \geq 0$, we obtain

$$p_e = \sum_{w=d}^{n} \sum_{k=0}^{t_1} \sum_{i=0}^{k} \sum_{\ell=0}^{k-\ell} N_w \frac{(n-w)!\, w!\, (q-1)^{-w+k-i-\ell} (q-2)^{i} p_o^{w+2\ell+i-k} (1-p_o)^{n-w-2\ell-i+k}}{\ell!\, (u-w-\ell)!\, i!\, (k-i-\ell)!\, (w-k-\ell)!}$$

Interchanging sums, substituting the upper bound of (B.9) for N_w, and writing the ranges of w, k, i and l more suggestively, we have

$$p_e \leq \sum_{k \leq t_1} \sum_{i \geq 0} \sum_{\ell \geq 0} \sum_{w \geq d} \frac{n!\, t_o!\, (n-w)!\, (q-1)^{k-\ell-i-t_o} (q-2)^{i} p_o^{w+2\ell+i-k} (1-p_o)^{n-w-2\ell-i+k}}{\ell!\, (n-w-\ell)!\, i!\, (k-\ell-i)!\, (w-k-\ell)!\, (n-w+t_o)!}$$

We now show that the dominant term in this expression is that specified by $k=t_1$, $i=0$, $l=0$, and $w=d$, and in fact that the whole series is bounded by

$$p_e \leq C_1C_2C_3C_4 \frac{n!\, t_o!\, (q-1)^{t_1-t_o} p_o^{d-t_1} (1-p_o)^{n-d+t_1}}{t_1!\, (d-t_1)!\, (n-d+t_o)!} \tag{B.10}$$

where

$$C_1 \equiv \frac{1}{1-a_1}, \qquad a_1 \equiv \frac{p_o}{1-p_o} \frac{n-d+t_o}{d-t_1+1}$$

$$C_2 \equiv \frac{1}{1-a_2}, \qquad a_2 \equiv \left(\frac{p_o}{1-p_o}\right)^{2} \frac{1}{q-1} \frac{(n-d)t_1}{d-t_1+1}$$

$$C_3 \equiv \frac{1}{1-a_3}, \quad a_3 \equiv \frac{p_o}{1-p_o}\,\frac{q-2}{q-1}\,t_1$$

$$C_4 \equiv \frac{1}{1-a_4}, \quad a_4 \equiv \frac{p_o}{1-p_o}\,\frac{1}{q-1}\,\frac{t_1}{d-t_1-1},$$

and it is assumed that the constants a_m are less than one. This result follows from repeated bounding of the series by the first term times a series of the form

$$\sum_{n\geq 0} a_m^n = \frac{1}{1-a_m}.$$

For example, the ratio of the $(w+1)^{st}$ to the w^{th} term is

$$\frac{p_o}{1-p_o}\,\frac{n-w-\ell}{n-w}\,\frac{n-w-t_o}{w-k+\ell+1} \leq a_1$$

since $w \geq d$, $k \leq t_1$, $l \geq 0$.

The ratio of the $(l+1)^{st}$ term to the l^{th} term is

$$\left(\frac{p_o}{1-p_o}\right)^2 \frac{1}{q-1}\,\frac{n-w-\ell}{\ell+1}\,\frac{k-\ell-i}{w-k+\ell+1} \leq a_2;$$

of the $(i+1)^{st}$ to the i^{th}:

$$\frac{p_o}{1-p_o}\,\frac{q-2}{q-1}\,\frac{k-\ell-i}{i+1} \leq a_3;$$

and of the $(k-1)^{st}$ to the k^{th}:

$$\frac{p_o}{1-p_o}\,\frac{1}{q-1}\,\frac{k-\ell-i}{w-k+\ell+1} \leq a_4.$$

The bound on p_e of Eq. B.10 is a valid upper bound, but not a good approximation, since (B.9) is a weak bound for N_w. A tighter bound would follow from better knowledge of N_w. In Table 5 we use the actual values of N_w for RS codes, which markedly affects the character of our results.

B.3 MODULATION ON A GAUSSIAN CHANNEL

We contemplate sending one of $M = 2^{R_o}$ biorthogonal signals over an infinite bandwidth additive white Gaussian noise channel. A well-known model[39] for such a transmission is this. The M signals are represented by the M $(M/2)$-dimensional vectors x_i, $1 \leq i \leq M/2$ or $-1 \geq i \geq -M/2$, which are the vectors with zeros in all places but the $|i|^{th}$, and in that place have $\pm L$ according to whether $i = \pm|i|$. (These vectors correspond to what would be observed at the outputs of the bank of M/2 matched filters if the

waveforms that they represent, uncorrupted by noise, were the input.)

The actual, noisy outputs of the bank of matched filters are represented by the $(M/2)$-dimensional vector $y = (y_1, y_2, \ldots, y_{M/2})$. If we assume a noise energy per dimension of N, then

$$\Pr(\vec{y}|\vec{x}_i) = \frac{1}{(2\pi N)^{M/2}} \exp - \sum_{j=1}^{M/2} \frac{(y_j - x_{ij})^2}{2N}.$$

Interpreting

$$\sum_{j=1}^{M/2} (y_j - x_{ij})^2$$

as the Euclidean distance between the vectors y and x_i, we see that the maximum-likelihood decision rule is to choose that input closest in Euclidean distance to the received signal.

The case $M=4$ is illustrated in Fig. B-1, where we have drawn in the lines marking the boundaries of the decision regions. There is perfect symmetry between the four inputs. If one of them, say $(L, 0)$, is selected, the probability of error is the probability that the received signal will lie outside the decision region that contains $(L, 0)$. If we let E_1 be the event that the received signal falls on the other side of the line AB from $(L, 0)$, and E_2 that it falls on the other side of CD, then it can readily be shown by a 45° coordinate rotation that E_1 and E_2 are independent, and that each has probability

$$p = \frac{1}{\sqrt{2\pi N}} \int_{L/\sqrt{2}}^{\infty} e^{-y^2/2N}\, dy$$

$$= \frac{1}{\sqrt{2\pi}} \int_{L/\sqrt{2N}}^{\infty} e^{-z^2/2}\, dz \equiv \Phi\left(\frac{L^2}{2N}\right).$$

The probability that neither occurs is $(1-p)^2$, so that the probability that at least one occurs, which is the probability of error, is

$$q = 2p - p^2.$$

When $M>4$, the symmetry between the inputs still obtains, so let us suppose the transmission of

$$\vec{x}_1 = (L, 0, \ldots, 0).$$

Let E_j, $2 \leq j \leq M/2$ be defined as the event in which the received signal is closer to one of the three vectors x_{-1}, x_j, x_{-j}, than to x_j. Then the event ϵ of an error is the union of these events

$$\epsilon = \bigcup_{j=2}^{M/2} E_j.$$

But the probability of any one of these events is q. Thus, by the union bound,

$$p_o = \Pr(\epsilon) \leq \sum_{j=2}^{M/2} \Pr(E_j) = \left(\frac{M}{2}-1\right) q. \qquad \text{(B. 11)}$$

When the signal-to-noise ratio L^2/N is large, the bound of Eqs. B. 7-B. 9 becomes quite tight. To calculate Φ, we use an approximation of Hastings.[40] Viterbi[41] has calculated the exact value of p for $3 \leq R \leq 10$; we have fitted curves to his data in the low signal-to-noise range, and used the bound above elsewhere, so that over the whole range p is given correctly within one per cent. When $R_o \geq 11$, the union bound is used for all

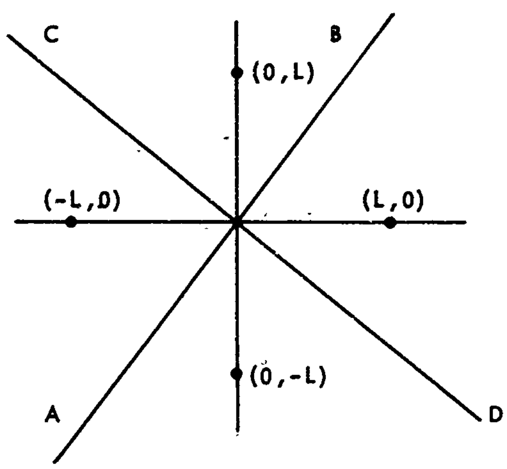

Fig. B-1. Illustrating the case M = 4.

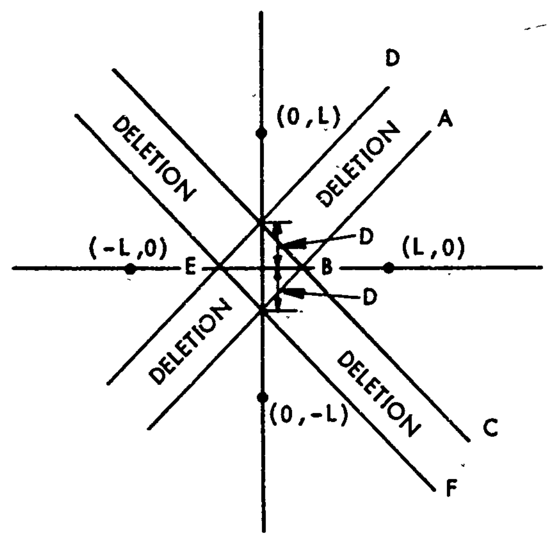

Fig. B-2. Decision and deletion regions (M=4).

signal-to-noise ratios.

Finally, we have the problem of bounding the deletion and error probabilities, when the detector deletes whenever the magnitude of the output of some matched filter is not at least D greater than that of any other. Figure B-2 illustrates the decision and deletion regions, again for M = 4. It is clear that the probability of not decoding correctly is computed exactly as before, with L replaced by L - D; this probability overbounds and approximates the deletion probability. The probability of error is overbounded, not tightly, by the probability of falling outside the shaded line DEF, which probability is computed as before with L replaced by L + D.

When M > 4, the union bound arguments presented above are still valid, again with L replaced by L - D for deletion probability and by L + D for error probability.

The case in which M = 2 is trivial.

Acknowledgment

Properly to acknowledge the help of all who contributed to such a work as this is next to impossible. I am grateful to the National Science Foundation and to Lincoln Laboratory, M. I. T., for their financial support during my graduate years. Intellectually, I perhaps owe most to the stimulating ambiance provided by fellow students, faculty members, and staff clustered around the Processing and Transmission of Information group in the Research Laboratory of Electronics, M. I. T., particularly my advisers, Professor John M. Wozencraft and Professor Robert G. Gallager. Professor Robert S. Kennedy was a helpful and responsive reader. Of those unseen hundreds whose written work contributed to my thinking, W. W. Peterson helped greatly with his timely and thorough exposition of coding theory. An absorbing course in thermodynamics given by Professor John A. Wheeler was probably responsible for my first interest in information theory, and a provocative seminar by Professor Claude E. Shannon for my return to the field for my doctoral work. The time-sharing facility of the Computation Center, M. I. T., very much expedited the search for the good codes in Section V of this report.

References

1. C. E. Shannon and W. Weaver, A Mathematical Theory of Communication (University of Illinois Press, Urbana, Ill., 1949); see also Bell System Tech. J. 27, 379 and 623 (1948).

2. J. M. Wozencraft and B. Reiffen, Sequential Decoding (The M. I. T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961).

3. J. L. Massey, Threshold Decoding (The M. I. T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1963).

4. W. W. Peterson, Error-Correcting Codes (The M. I. T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961).

5. R. G. Gallager, "A Simple Derivation of the Coding Theorem and Some Applications," IEEE Trans., Vol. IT-11, p. 1, 1965.

6. D. Slepian, "A Class of Binary Signalling Alphabets," Bell System Tech. J. 35, 203 (1956).

7. P. Elias, "Coding for Noisy Channels," IRE Convention Record, Part 4, p. 37, 1955; see also W. W. Peterson, op. cit., Chapter 12.

8. I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," J. SIAM 8, 300 (1960).

9. J. M. Wozencraft and M. Horstein, "Coding for Two-Way Channels," Information Theory (Fourth London Symposium), C. Cherry (ed.) (Butterworths, Washington, 1961), p. 11.

10. W. W. Peterson, op. cit., Section 4.6 and Chapter 10.

11. R. W. Hamming, "Error-Detecting and Error-Correcting Codes," Bell System Tech J. 29, 147 (1950).

12. E. Prange, "The Use of Information Sets in Decoding Cyclic Codes," (Brussels Symposium), IRE Trans., Vol. IT-8, p. 5, 1962.

13. T. Kasami, "A Decoding Procedure for Multiple-Error-Correcting Cyclic Codes," IRE Trans., Vol. IT-10, p. 134, 1964.

14. J. MacWilliams, "Permutation Decoding of Systematic Codes," Bell System Tech. J. 43, 485 (1964).

15. L. D. Rudolph and M. E. Mitchell, "Implementation of Decoders for Cyclic Codes," IEEE Trans., Vol. IT-10, p. 259, 1964.

16. P. Elias, "Coding for Two Noisy Channels," Information Theory (Third London Symposium), C. Cherry (ed.) (Academic Press, New York, 1956), p. 61.

17. H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," Ann. Math. Stat. 23 (1952).

18. R. G. Gallager, Private communication (course notes), 1963.

19. C. E. Shannon and R. G. Gallager, Private communications, 1963.

20. E. N. Gilbert, "A Comparison of Signalling Alphabets," Bell System Tech. J. 31, 504 (1952).

21. A. A. Albert, Fundamental Concepts of Higher Algebra (University of Chicago Press, Chicago, Ill., 1956).

22. R. C. Singleton, "Maximum Distance q-Nary Codes," IEEE Trans., Vol. IT-10, p. 116, 1964.

23. I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," J. SIAM 8, 300 (1960). As N. Zierler has pointed out, these codes are most easily understood and implemented as BCH codes (cf. W. W. Peterson, op. cit., Section 9.3).

24. R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," Inform. Contr. 3, 68 (1960).

25. A. Hocquenghem, "Codes Correcteurs d'Erreurs," Chiffres 2, 147 (1959).

26. D. Gorenstein and N. Zierler, "A Class of Cyclic Linear Error-Correcting Codes in p^m Symbols," J. SIAM 9, 207 (1961); see W. W. Peterson, op. cit., Section 9.4.

27. G. D. Forney, Jr., "Decoding Bose-Chaudhuri-Hocquenghem Codes," Quarterly Progress Report No. 76, Research Laboratory of Electronics, M.I.T., Cambridge, Mass., January 15, 1965, pp. 232-240.

28. D. D. McCracken and W. S. Dorn, Numerical Methods and Fortran Programming (John Wiley and Sons, Inc., New York, 1964), Chapter 8.

29. W. W. Peterson, Error-Correcting Codes, op. cit., Chapter 7.

30. T. C. Bartee and D. I. Schneider, "Computation with Finite Fields," Inform. Contr. 6, 79 (1963).

31. R. T. Chien, "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes," IEEE Trans., Vol. IT-10, p. 357, 1964.

32. T. C. Bartee and D. I. Schneider, "An Electronic Decoder for Bose-Chaudhuri-Hocquenghem Error-Correcting Codes" (Brussels Symposium), IRE Trans., Vol. IT-8, p. 17, 1962.

33. W. W. Peterson, "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," IRE Trans., Vol. IT-6, p. 459, 1960.

34. N. Zierler, "Project 950.9: Error-Correcting Coder-Decoder. Summary of Results, Phase 1," TM 4109, MITRE Corporation, Bedford, Mass., 29 October 1964.

35. J. E. Savage, "The Computation Problem with Sequential Decoding," Technical Report 439, Research Laboratory of Electronics, M.I.T., Cambridge, Mass., February 16, 1965 (also Lincoln Laboratory Technical Report 371).

36. P. Elias, "Error-Free Coding," IRE Trans., Vol. PGIT-4, p. 29, 1954.

37. J. Ziv, Private communication, 1964 (unpublished paper).

38. G. H. Hardy, J. E. Littlewood, and G. Pólya, Inequalities (Cambridge University Press, London, 1952), Chapter 2.

39. S. W. Golomb (ed.), Digital Communications with Space Applications (Prentice-Hall, Englewood Cliffs, N.J., 1964).

40. D. Hastings, Approximations for Digital Computers (Princeton University Press, Princeton, N.J., 1955).

41. A. Viterbi, in S. W. Golomb (ed.), op. cit., Appendix 4.

JOINT SERVICES DISTRIBUTION LIST

Department of Defense

Defense Documentation Center
Attn: TISIA
Cameron Station, Bldg. 5
Alexandria, Virginia 22314

Director, National Security Agency
Attn: C3/TDL
Fort George G. Meade, Maryland 20755

Mr. Charles Yost, Director
For Materials Sciences
Advanced Research Projects Agency, DOD
Washington, D.C. 20301

Director
Advanced Research Projects Agency
Department of Defense
Washington, D.C. 20301

Dr. James A. Ward
Office of Deputy Director (Research and Information Rm. 3D1037) DOD
The Pentagon
Washington, D.C. 20301

Dr. Edward M. Reilley
Asst. Director (Research)
Ofc of Defense Res. & Eng., DOD
Washington, D.C. 20301

Department of the Army

Librarian PTA130
United States Military Academy
West Point, New York 10996

Director
U.S. Army Electronics Laboratories
Fort Monmouth, New Jersey 07703
Attn: AMSEL-RD-ADT

	NP	SE
DR	NR	SR
FU#1	PE	SS
GF	PF	X
NE	PR	XC
NO	SA	XE
		XS

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-SC
Fort Monmouth, New Jersey 07703

C.O., Harry Diamond Laboratories
Attn: Mr. Berthold Altman
Connecticut Ave. & Van Ness St. N.W.
Washington, D.C. 20438

The Walter Reed Institute of Research
Walter Reed Army Medical Center
Washington, D.C. 20012

Director
U.S. Army Electronics Laboratories
Attn: Mr. Robert O. Parker, Executive Secretary, JSTAC (AMSEL-RD-X)
Fort Monmouth, New Jersey 07703

Director
U.S. Army Electronics Laboratories
Attn: Dr. S. Benedict Levin, Director Institute of Exploratory Research
Fort Monmouth, New Jersey 07703

Commanding Officer
U.S. Army Research Office (Durham)
Attn: CRD-AA-IP (Richard O. Ulsh)
P.O. Box CM, Duke Station
Durham, North Carolina 27706

Commanding Officer
U.S. Army Medical Research Laboratory
Fort Knox, Kentucky

Commanding Officer
U.S. Army Personnel Research Office
Washington, D.C.

Dr. H. Robl, Deputy Director
U.S. Army Research Office (Durham)
P.O. Box CM, Duke Station
Durham, North Carolina 27706

Commandant
U.S. Command and General Staff College
Attn: Secretary
Fort Leavenworth, Kansas 66207

Director
U.S. Army Eng. Geodesy, Intell. and Mapping
Research & Development Agcy.
Fort Belvoir, Virginia 22060

Commanding Officer
Human Engineering Laboratories
Aberdeen Proving Ground, Maryland 21005

Commanding Officer
U.S. Limited War Laboratory
Attn: Technical Director
Aberdeen Proving Ground, Maryland 21005

Commanding Officer
U.S. Army Security Agency
Arlington Hall, Arlington, Virginia 22212